

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

(повна назва інституту/факультету)

Автоматизованих систем обробки інформації і управління

(повна назва кафедри)

«На правах рукопису»

УДК 004.41

До захисту допущено

В.о. завідувача кафедри

Олександр ПАВЛОВ

"__" _____ 2020 р.

Магістерська дисертація

на здобуття ступеня магістра

**за освітньо-науковою програмою «Інженерія програмного
забезпечення комп'ютеризованих систем»**

зі спеціальності 121 «Інженерія програмного забезпечення»

**на тему: «Методи та засоби інтеграції спеціалізованого торгівельного
обладнання з метою автоматизації бізнес процесів підприємства торгівлі»**

Виконав:

студент VI курсу, групи ПІ-81мн

Педос Василь Миколайович

Керівник:

Кандидат технічних наук, доцент

Ліщук Катерина Ігорівна

Рецензент:

Кандидат технічних наук, доцент

Лісовиченко Олег Іванович

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Автоматизованих систем обробки інформації і управління

Рівень вищої освіти – другий (магістерський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-наукова програма – «Інженерія програмного забезпечення комп'ютеризованих систем»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ

" ____ " _____ 2020 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Педосу Василю Миколайовичу

1. Тема дисертації "Методи та засоби інтеграції спеціалізованого торговельного обладнання з метою автоматизації бізнес процесів підприємства торгівлі", науковий керівник дисертації Ліщук Катерина Ігорівна, к.т.н., доц., затверджені наказом по університету від " 24 " березня 2020 року № 910-с
2. Термін подання студентом дисертації 27 квітня 2020
3. Об'єкт дослідження Інформаційні системи автоматизації обліку й керування підприємством
4. Предмет дослідження Засоби автоматизації складських та торгових процесів шляхом інтеграції спеціалізованого торговельного обладнання до існуючої інформаційної системи підприємства
5. Перелік завдань, які потрібно розробити Аналіз програмно-технічних засобів інтеграції спеціалізованого торговельного обладнання; аналіз структури типової інформаційної системи підприємства торгівлі; розробка уніфікованого механізму інтеграції торговельного обладнання до існуючої інформаційної мережі підприємства; проектування та розробка прикладного рішення на основі розробленого уніфікованого механізму.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу: Концептуальна архітектура сервера автоматизації; Структурна схема компонентів сервера автоматизації

7. Орієнтовний перелік публікацій: Механізм інтеграції спеціалізованого обладнання до інформаційної системи підприємства торгівлі; Уніфікований механізм інтеграції спеціалізованого торговельного обладнання до інформаційної системи на базі платформи "ІС: Підприємство 8"; Система автоматизації складських та торгових процесів

8. Консультанти розділів дисертації

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|--------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| | | | |
| | | | |
| | | | |
| | | | |

9. Дата видачі завдання _____

Календарний план

| № з/п | Назва етапів виконання магістерської дисертації | Строк виконання етапів магістерської дисертації | Примітка |
|-------|--|---|----------|
| 1 | Вивчення та аналіз предметної області | 29.11.2018 | |
| 2 | Порівняльний аналіз програмно-технічних засобів інтеграції спеціалізованого торговельного обладнання | 14.02.2019 | |
| 3 | Аналіз структури типової інформаційної системи підприємства торгівлі | 24.04.2019 | |
| 4 | Проектування уніфікованого механізму інтеграції торговельного обладнання | 08.10.2019 | |
| 5 | Проектування та розробка прикладного рішення | 10.02.2020 | |
| 6 | Оформлення документації | 18.04.2020 | |
| 7 | Подання роботи на попередній захист | 23.04.2020 | |
| 9 | Подання роботи на рецензію | 10.05.2020 | |
| 10 | Подання роботи на основний захист | 18.05.2020 | |

Студент

Василь ПЕДОС

Науковий керівник

Катерина ЛІЩУК

РЕФЕРАТ

Магістерська дисертація складається зі вступу, чотирьох розділів, висновку та 2 додатків, містить 25 рисунків і 24 таблиць. Повний обсяг магістерської дисертації складає 106 сторінок, з яких додатки займають 16 сторінок.

Актуальність теми. На сьогоднішній день автоматизація ключових бізнес-процесів підприємства є актуальним і важливим напрямком досліджень у сфері теорії та практики управління. Однак іноді виникає необхідність не лише автоматизувати процес, використовуючи програмні засоби, а й задіяти спеціалізоване обладнання.

Аналіз існуючих програмних та технічних рішень показав, що сучасні програмні комплекси, призначені для забезпечення взаємодії платформи "1С:Підприємство 8" (що є фактичним монополістом у сфері автоматизації бізнес-процесів торговельних підприємств на території країн СНД та на українському ринку зокрема) зі сторонніми пристроями або потребують внесення змін до конфігурації платформи, або ж накладають жорсткі обмеження на сферу їх застосування.

Тому постало питання розробки універсального механізму інтеграції спеціалізованого торговельного обладнання з платформою "1С:Підприємство 8" з метою автоматизації складських та торгових операцій, що дозволило б мінімізувати ризики та знизити вплив інтеграції стороннього торговельного обладнання на стабільність існуючої інформаційної системи підприємства торгівлі.

Мета дослідження. Метою дослідження стала розробка універсального механізму інтеграції спеціалізованого обладнання до інформаційної системи підприємства торгівлі, а також створення прикладної програмної реалізації на його основі.

Для реалізації поставленої мети були сформульовані **наступні завдання:**

- проаналізувати існуючі програмні та технічні засоби інтеграції

спеціалізованого торгівельного обладнання;

- дослідити та проаналізувати структуру типової інформаційної системи підприємства торгівлі на базі платформи "1С:Підприємство 8";

- розробити уніфікований механізм інтеграції торгівельного обладнання до існуючої інформаційної мережі підприємства;

- спроектувати прикладне рішення для вирішення задачі автоматизації бізнес-процесу товарообігу із залученням терміналів збору даних та дисплеїв покупця на основі розробленого уніфікованого механізму;

- розробити спроектоване програмне забезпечення.

Об'єктом дослідження є інформаційні системи автоматизації обліку й керування підприємством.

Предметом дослідження є засоби автоматизації складських та торгових процесів шляхом інтеграції спеціалізованого торгівельного обладнання до існуючої інформаційної системи підприємства.

Методи дослідження. Поставлені завдання дослідження виконувалося із застосуванням наступних методів:

- в процесі дослідження предметної області та існуючих програмно-технічні засобів вирішення задачі інтеграції торгівельного обладнання було застосовано методи узагальнення, класифікації та порівняння;

- з метою практичної перевірки в дії теоретично описаного уніфікованого механізму інтеграції спеціалізованого обладнання застосовувався емпіричний метод експерименту;

- для формалізації бізнес-процесу інвентаризації на торгівельному підприємстві було застосовано методи узагальнення та декомпозиції.

Наукова новизна одержаних результатів дослідження полягає у створенні універсального механізму міжсистемної взаємодії інформаційної системи на базі платформи "1С: Підприємство 8" та торгівельного обладнання, що дозволяє мінімізувати ризики такої інтеграції за рахунок уникнення необхідності внесення змін до поточної конфігурації 1С.

Практичне значення отриманих результатів полягає в мінімізації ризиків автоматизації бізнес-процесів підприємства засобами впровадження

стороннього спеціалізованого торгівельного обладнання до інформаційного комплексу, побудованого на базі платформи "1С: Підприємство 8".

Зв'язок роботи з науковими програмами, планами, темами.

Дисертаційна робота магістра виконувалась у Національному технічному університеті України "Київський політехнічний інститут ім. Ігоря Сікорського" згідно з планом науково-дослідницьких робіт кафедри автоматизованих систем обробки інформації та управління.

Публікації. Наукові положення дисертації були опубліковані в тезах IV Всеукраїнської науково-практичної конференції молодих вчених та студентів "Інформаційні системи та технології управління" (ІСТУ-2020) та в тезах Міжнародної наукової інтернет-конференції "Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення" (випуски 30 та 48).

Ключові слова. АВТОМАТИЗАЦІЯ, ТОРГІВЕЛЬНЕ ОБЛАДНАННЯ, ІНТЕГРАЦІЯ, УНІФІКОВАНИЙ МЕХАНІЗМ, СКЛАДСЬКИЙ ОБЛІК, ТОРГОВИЙ ОБЛІК

ABSTRACT

The master's thesis consists of an introduction, four chapters, a conclusion and 2 appendices, contains 25 figures and 24 tables. Full range of master's thesis is 106 pages, including 16 pages of appendices.

Topicality of theme. Nowadays automation of key enterprise business processes is a relevant and important area of research in the field of management theory and practice. However, sometimes it is necessary not only to automate the process using software, but also to use specialized hardware equipment.

Analysis of existing software and technical solutions discovered that modern software packages designed for integration "1C: Enterprise 8" platform (which is a de facto monopolist in the field of commercial enterprises business processes automation on CIS and Ukrainian market) with third-party devices either require changes in the platform configuration or impose severe restrictions on their scope.

Therefore, the question of developing a universal mechanism for the specialized trading equipment integration with "1C: Enterprise 8" platform to automate warehousing and trading operations, which would minimize risks and reduce the impact of third-party trading equipment integration on the stability of the existing information system.

The aim of the research. The aim of the research is to develop a universal mechanism for specialized equipment integration into trade enterprise information system and to create a software solution based on it.

To achieve the aim, **the following objectives** were set:

- to analyze an existing specialized trade equipment integration software and hardware means;
- to investigate and analyze a typical trade enterprise information system structure based on "1C: Enterprise 8" platform;
- to develop a unified mechanism of trade equipment integration into the existing enterprise information system;
- to design a software solution for trade business processes automation with

data collection terminals and informational displays involvement based on the developed unified mechanism;

- develop designed software solution.

The object of the research is accounting and enterprise management automation systems.

The subject of the research is warehousing and trading processes automation means based on specialized trade equipment integrating into the existing enterprise information system.

Methods of the research. The objectives of the research were performed using the following methods:

- during the subject area research and existing software and hardware solutions for the trade equipment integration problem investigation, methods of generalization, classification and comparison were applied;
- in the scope of practical verification of the theoretically described specialized equipment integration mechanism the empirical method of experiment was applied;
- generalization and decomposition methods were applied to formalize trading enterprise inventory business process.

Scientific novelty of the results is to create the universal intersystem interaction mechanism of software solutions based on "1C: Enterprise 8" platform and trade equipment, which minimizes the risks of such integration by avoiding the needs of changes in the current 1C configuration.

The practical significance of the results is to minimize the risks of enterprise business processes automation by means of third-party specialized trade equipment integration into the software system based on "1C: Enterprise 8" platform.

Relationship with scientific programs, plans, themes. The master's thesis was implemented at the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" according to the research works plan of the automated information processing and control systems department.

Publications. The scientific provisions of the thesis were published in the thesis of the IV All-Ukrainian scientific-practical conference of young scientists and students "Information systems and management technologies" (ISMT-2020) and in

the theses of the International scientific internet-conference "Information society: technological, economic and technical aspects of formation" (issues 30 and 48).

Keywords. AUTOMATION, TRADE EQUIPMENT, INTEGRATION, UNIFIED MECHANISM, WAREHOUSE ACCOUNTING, TRADE ACCOUNTING

ЗМІСТ

| | |
|--|----|
| Вступ..... | 12 |
| 1 Задача інтеграції спеціалізованого торговельного обладнання | 14 |
| 1.1 Аналіз предметної області | 15 |
| 1.2 Програмно-технічні засоби інтеграції торговельного обладнання..... | 17 |
| 1.2.1 Вбудовані засоби інтеграції | 18 |
| 1.2.2 Засоби міжсистемної взаємодії | 20 |
| 1.3 Огляд існуючих програмних рішень..... | 23 |
| 1.4 Постановка задачі інтеграції торговельного обладнання | 25 |
| Висновки до розділу..... | 26 |
| 2 Методи та засоби побудови уніфікованої інтеграційної платформи | 28 |
| 2.1 Опис інформаційної системи підприємства..... | 28 |
| 2.2 Узагальнений механізм інтеграції торговельного обладнання | 30 |
| 2.3 Концептуальна модель сервера автоматизації | 32 |
| 2.3.1 Засоби конфігурації сервера | 35 |
| 2.3.2 Багатокористувацький режим роботи..... | 36 |
| 2.3.3 Міжмережева клієнт-серверна взаємодія | 37 |
| Висновки до розділу..... | 37 |
| 3 Програмна реалізація прикладного рішення | 38 |
| 3.1 Структурне моделювання програмного забезпечення | 38 |
| 3.1.1 Адміністрування інтеграційної системи | 38 |
| 3.1.2 Процеси товарообігу в контексті роздрібної торгівлі | 40 |
| 3.2 Опис програмної реалізації | 43 |
| 3.2.1 Сервісно-орієнтована будова програмного рішення | 45 |

| | |
|---|-----|
| 3.2.2 Клієнтські мобільні застосунки | 48 |
| 3.2.3. Міжмережева взаємодія компонентів системи | 51 |
| 3.2.4 Архітектура WEB-серверу | 63 |
| 3.3 Аналіз безпеки даних | 64 |
| Висновки до розділу | 65 |
| 4 Впровадження та супровід програмного забезпечення | 66 |
| 4.1 Розгортання програмного забезпечення | 66 |
| 4.2 Експлуатація програмного забезпечення | 68 |
| 4.2.1 Керування сервером автоматизації | 68 |
| 4.2.2 Сервіс "Інвентаризація" | 72 |
| 4.2.3 Сервіс "Дисплей покупця" | 81 |
| Висновки до розділу | 85 |
| Висновки | 86 |
| Перелік посилань | 87 |
| Додаток А. Тексти програмного коду | 89 |
| Додаток Б. Графічний матеріал | 104 |

ВСТУП

Ускладнення структури підприємства, його розширення, збільшення чисельності працівників, зростання кількості інформаційних процесів – всі ці фактори суттєво впливають на спроможність підприємства до адаптації в нових реаліях ведення бізнесу. В той самий час у випадку реального збільшення обсягів інформації та ускладнення бізнес-процесів існуючий штат керуючого персоналу та аналітиків не завжди спроможний тримати під контролем усі аспекти управління підприємством, що підвищує ризики затримки та навіть ставить під сумнів можливість реалізації окремих проектів. Якщо до певної межі компанія має можливості та ресурси керувати бізнес-процесами у ручному режимі, то з часом виникає реальна проблема збереження контролю над підприємством в цілому. За такої ситуації у підприємства є два варіанти виходу з кризової ситуації: збільшення штату фахівців або автоматизація бізнес-процесів.

Суть автоматизації полягає у підвищенні ефективності підприємства – у тому числі й економічної – за рахунок введення додаткової ланки апаратного-програмного забезпечення. Це дозволяє за рахунок мінімізації впливу так званого "людського фактору" зменшити ризики порушення ключових бізнес-процесів підприємства та покращити його функціонування в цілому. Слід пам'ятати, що автоматизація бізнес-процесів не ставить на меті повністю виключити людину з діяльності підприємства, а лише мінімізує її роль у виконанні операційних задач, за рахунок чого збільшується ефективність розподілу навантаження між машиною та людиною.

Одним з лідерів та фактичним монополістом у сфері автоматизації бізнес-процесів торгівельного підприємства на території СНД є компанія 1С зі своїм продуктом "1С:Підприємство 8".

Однак автоматизація бізнес-процесів підприємства торгівлі не завжди можлива лише у площині програмного рішення, але і часто потребує інтеграції спеціалізованого обладнання. Платформа "1С: Підприємство 8" має вбудовані

засоби взаємодії лише з обмеженим набором торговельного обладнання – так званими офіційно сумісними пристроями. Але у разі необхідності використання стороннього обладнання виникає потреба у внесенні змін до існуючої конфігурації інформаційної системи підприємства, що може негативно вплинути на її стабільність, а у довгостроковій перспективі призвести до порушення ключових бізнес-процесів підприємства. Саме тому досить гостро постає питання створення уніфікованого механізму інтеграції різноманітного обладнання до інформаційної системи підприємства торгівлі, що дозволив би організувати процес взаємодії зі сторонніми пристроями, мінімізуючи при цьому необхідність модифікації поточної конфігурації платформи "1С: Підприємство 8", що і стало основним завданням даної роботи.

Мета дослідження полягає у мінімізації ризиків та зменшенні впливу інтеграції стороннього торговельного обладнання на стабільність існуючої інформаційної системи підприємства за рахунок впровадження уніфікованого механізму інтеграції, що дозволив би зменшити зв'язність компонентів в цілому, спростити процес підтримки та розширення підприємства, а також звести до мінімуму вплив людського фактору на стратегічні бізнес-процеси підприємства.

1 ЗАДАЧА ІНТЕГРАЦІЇ СПЕЦІАЛІЗОВАНОГО ТОРГІВЕЛЬНОГО ОБЛАДНАННЯ

В умовах мінливого ринкового середовища кожне підприємство прагне відшукати як зовнішні так і внутрішні резерви, що дозволили б підвищити ефективність власної діяльності та мінімізувати витрати виробництва, а також забезпечити вигідне конкурентоспроможне становище у власному діловому економічному середовищі. На підґрунті як внутрішньодержавної, так і загальносвітової економічної нестабільності набуває актуальності питання розвитку внутрішньокорпоративних технологій управління, що не лише підтримують життєдіяльність бізнесу, а й сприяють оптимізації внутрішніх процесів, скорочуючи рівень задіяності людини.

На сьогоднішній день автоматизоване управління бізнес-процесами стало не просто важливою, а невід'ємною складовою ефективного ведення бізнесу. У загальному випадку автоматизація бізнес-процесів полягає у впровадженні спеціалізованого апаратно-програмного забезпечення, функціонування якого спрямоване на пришвидшення та спрощення прийняття управлінських рішень, а також покращення якості виконання типових задач підприємства. Ключова ідея використання цифрових інструментів автоматизації полягає у ефективному керуванні та операціюванні великими обсягами інформації, завданнями, послугами та товарами компанії.

Автоматизація бізнес-процесів також спрямована на вирішення задачі нейтралізації впливу "людського фактору" на стабільність функціонування ключових процесів підприємства. Проте основною метою автоматизації є не повноцінна заміна співробітників апаратно-програмними комплексами, що функціонують без втручання людини. Навпаки, автоматизація спрямована на позбавлення персоналу від рутинних та механічних завдань, що дозволяє сфокусуватися на виконанні нових обов'язків, дослідницьких завдань або іншої діяльності, що передбачає застосування особистого досвіду та специфічних знань співробітника.

Головним призначення автоматизації бізнес-процесів є підвищення якості виконання самого процесу. Автоматизованому процесу притаманні більш стабільні характеристики, ніж процесу, що виконується у ручному режимі співробітниками компанії. У більшості випадків автоматизації сприяє підвищенню продуктивності, скороченню часу виконання процесу, зменшенню ризику виникнення помилок, зниженню вартості, а також збільшенню точності та стабільності виконуваних операцій.

Автоматизація бізнес-процесів підприємства незалежно від галузі діяльності, географічного розташування, особливостей ринку та напрямків розвитку компанії, виступає основою для формування принципово нового підходу до ведення підприємницької діяльності та прийняття управлінських рішень. У сучасних умовах цифрової економіки компанії здатні прискорити свій розвиток, підвищити ефективність управління, поліпшити показники якості обслуговування та знизити витрати виробництва за рахунок інтелектуальних досягнень людства в області автоматизації бізнес-процесів.

1.1 Аналіз предметної області

Задача автоматизації бізнес-процесів вирішується засобами впровадження спеціалізованих корпоративних інформаційних систем. Функціонування таких систем спрямоване на оптимізацію керування трудовими та матеріальними ресурсами підприємства, спрощення фінансового менеджменту, організацію ефективного управління активами тощо.

У загальному випадку корпоративні інформаційні системи автоматизації побудовані на багаторівневій модульній архітектурі.

В основі подібних систем лежить рівень збереження та управління даними, що відповідає за організацію сховища даних та забезпечення доступу до інформації.

Наступним рівнем є програмна платформа, що надає базові інструменти для обробки, інтерпретації та перетворення даних, а також відповідає за створення середовища для роботи як внутрішніх, так і зовнішніх модулів та

компонентів вищого рівня. Платформа у свою чергу складається з ядра – програмного контексту виконання зовнішніх модулів, та базового функціоналу, що вбудований в платформу та є її невід'ємною складовою. Базовий функціонал містить ключові моделі даних, необхідні для забезпечення життєдіяльності системи в цілому.

Програмна платформа надає інтерфейс для інтеграції сторонніх модулів – компонентів, що можуть під'єднуватися за необхідністю. Такі компоненти виконуються в ізольованих контекстах незалежно одна від одної та відповідають за функціонування окремих складових бізнес-логіки діяльності підприємства. Зовнішні модулі отримують доступ до єдиної бази даних системи та базового функціоналу платформи.

У сучасних ринкових умовах одним із лідерів та фактичним монополістом серед постачальників корпоративних інформаційних систем автоматизації бізнес-процесів підприємства як в країнах СНД, так і в Україні зокрема, є компанія 1С з продуктами сімейства "1С:Підприємство 8" [8].

Комплекс програм "1С: Підприємство" складається з технологічної платформи та розроблених на її основі прикладних рішень – конфігурацій – що призначені для вирішення конкретних задач підприємства. Така архітектура забезпечує відкритість прикладних рішень, спрощення їх розширення та підтримки, гнучкість конфігурації окремих компонентів системи, короткі терміни впровадження та масштабованість.

Гнучкість платформи дозволяє застосовування "1С: Підприємство 8" для вирішення широкого класу завдань автоматизації незалежно від галузі діяльності підприємства. До ключових функціональних можливостей належать:

- автоматизація виробничих та торгових підприємств, бюджетних і фінансових організацій, підприємств сфери обслуговування тощо;
- підтримка оперативного управління підприємством;
- автоматизація організаційної та господарської діяльності;
- ведення бухгалтерського обліку з регламентованою звітністю;
- широкі можливості для ведення управлінського обліку та побудови аналітичної звітності, зокрема підтримка багатовалютного обліку;

- вирішення задач планування, бюджетування та фінансового аналізу;
- задачі розрахунку заробітної плати та управління персоналом.

Проте зазвичай для ефективного вирішення задач автоматизації необхідне впровадження не лише програмного, а й додаткового апаратного забезпечення. На різних етапах діяльності підприємства може бути доцільним залучення різноманітного спеціалізованого обладнання, такого як термінали збору даних, принтери етикеток, сканери штрих-кодів, інформаційні дисплеї тощо.

Програмні продукти компанії 1С мають вбудовані засоби роботи з обмеженим набором спеціалізованого обладнання, що входить до переліку офіційно підтримуваного платформою. Але часто виникає необхідність впровадження стороннього обладнання, що офіційно не підтримується компанією 1С. У загальному випадку така інтеграція виконується шляхом модифікації типових прикладних конфігурацій платформи. Проте такий підхід потенційно може негативно вплинути на процес підтримки програмного комплексу, створити додаткові ризики або і зовсім унеможливити автоматичні оновлення конфігурації платформи, а також дестабілізувати інформаційний комплекс та порушити функціонування ключових бізнес-процесів підприємства в цілому.

Саме тому гостро постає питання створення уніфікованого механізму інтеграції спеціалізованого обладнання до інформаційної системи на базі платформи "1С: Підприємство 8", що дозволив би мінімізувати необхідність внесення змін до існуючих конфігурацій з одного боку, та забезпечив би тісну взаємодію з об'єктами та моделями даних платформи з іншого.

1.2 Програмно-технічні засоби інтеграції торгівельного обладнання

Програмний комплекс "1С: Підприємство 8" призначений для вирішення найрізноманітніших завдань автоматизації діяльності організацій. Платформа 1С з одного боку має потужні засоби конфігурації, які дозволяють налаштувати систему у відповідності до особливостей обробки інформації на конкретному

підприємстві, а з іншого є відкритою системою, та надає широкі можливості для інтеграції із зовнішнім програмним забезпеченням та різноманітним спеціалізованим обладнанням базуючись на відкритих загальноприйнятих протоколах міжсистемної взаємодії.

1.2.1 Вбудовані засоби інтеграції

Платформа "1С: Підприємство 8" надає гнучкі вбудовані засоби для інтеграції типового торговельного обладнання до інформаційної системи підприємства. Найпопулярнішими з таких механізмів вважаються інтеграція з застосуванням сертифікованих драйверів зовнішнього обладнання, керовані форми та використання мобільної платформи 1С [18].

1.2.1.1 Драйвер зовнішнього обладнання

Механізм інтеграції заснований на застосуванні сертифікованих драйверів зовнішнього обладнання покликаний спростити процес впровадження такого типового обладнання, як сканери штрих-кодів, електронні ваги, зчитувачі магнітних карт, тощо [9].

Даний підхід передбачає надання виробником торговельного обладнання спеціальної зовнішньої компоненти – драйвера роботи з обладнанням. Драйвер з одного боку реалізує інтерфейс обміну даними та повідомленнями з платформою 1С, а з іншого відповідальний безпосередньо за взаємодію із торговельним обладнанням.

Недоліками механізму є жорсткі обмеження на типи підтримуваних пристроїв з боку платформи 1С, а також додаткова залежність від виробників обладнання.

1.2.1.2 Керовані форми

Технологія керованих форм передбачає розділення бізнес-логіки елементу керування та його відображення між серверним та клієнтським модулями. Ключовими особливостями такого підходу є:

- декларативний опис інтерфейсу, тобто описується загальна структура елементів керування, а їх конкретне розміщення відбувається автоматично під час відображення;

- функціонал форми описується в термінах реквізитів та команд, де реквізити – це дані, з яким працює форма, а команди – це доступні дії;
- бізнес-логіка форми виконується виключно на віддаленому сервері, а клієнтський модуль відповідальний лише за відображення поточного стану та ввід даних.

Важливою особливістю керованих форм є можливість виконання клієнтського модуля як на "товстих" та "тонких" клієнтах 1С, так і на веб-клієнті 1С, що фактично знімає обмеження з операційної системи клієнтського пристрою.

В загальному випадку дана технологія може бути успішно застосована для інтеграції різноманітних мобільних пристроїв, планшетів чи дисплеїв до інформаційної системи підприємства, проте вона надає обмежені можливості у використанні специфічних функцій торговельного обладнання, таких як, наприклад, сканування RFID міток.

1.2.1.3 Мобільна платформа

Мобільна платформа 1С призначена для розробки та налагодження мобільних додатків для пристроїв під керуванням операційних систем iOS, Android та Windows.

Існує три варіанти застосування мобільної платформи, кожен з яких передбачає різні можливості взаємодії з інформаційною базою 1С:

- мобільний клієнт, що дозволяє взаємодіяти з інформаційною базою виключно в онлайн режимі, аналогічно тому, як це роблять клієнтські застосунки для настільних комп'ютерів та веб-клієнт;
- мобільний клієнт з автономним режимом, що в залежності від наявності мережевого з'єднання дозволяє взаємодіяти з інформаційною базою 1С як онлайн, так і з використанням локальної копії інформаційної бази на мобільному пристрої (даний режим роботи доступний починаючи з версії платформи 1С 8.3.16);
- самостійна мобільна платформа, яка використовує для роботи виключно локальну інформаційну базу на мобільному пристрої та призначена для повноцінного автономного функціонування.

Проте обмеження, що мобільна платформа 1С накладає на операційну систему, під керуванням якої функціонує обладнання, унеможливають інтеграцію широкого класу спеціалізованих торговельних пристроїв.

1.2.2 Засоби міжсистемної взаємодії

Вирішення задачі інтеграції нетипового програмного забезпечення чи спеціалізованого обладнання з програмним комплексом на базі платформи "1С: Підприємство" у загальному випадку здійснюється з використанням базових механізмів побудови міжсистемної взаємодії:

- застосування механізму WEB-сервісів;
- використання зовнішніх компонент 1С на основі NativeAPI;
- взаємодія з 1С на основі COM технології;
- використання "1С:Підприємство 8" у якості OLE Automation-сервера.

1.2.2.1 Механізм WEB-сервісів

WEB-сервіс – це мережева технологія, що забезпечує міжпрограмну взаємодію (між різними додатками) на основі веб-стандартів. WEB-сервіси забезпечують можливість "звернення" з одного додатку до іншого з метою запуску певних процесів чи отримання даних [20]. За даного підходу передача даних виконується у форматі XML (Extensible Markup Language) та протоколів, заснованих на його основі, таких як, наприклад, SOAP (Simple Object Access Protocol), що може використовуватися з будь-яким протоколом прикладного рівня: SMTP, FTP, HTTPS тощо, але найчастіше SOAP використовується поверх HTTP.

Також SOAP використовується для реалізації механізму віддаленого виклику процедур (RPC) [4].

Дана технологія тісно інтегрована до платформи "1С: Підприємство 8". Завдяки цьому 1С може виконувати роль "серверу додатків" – постачальника сервісів, що дозволяє організувати взаємодію між програмним комплексом 1С, та додатками, розробленими на інших платформах.

Проте критичним недоліком механізму використання WEB-сервісів є необхідність постійної модифікації конфігурації платформи 1С, що може

негативно впливати на стабільність функціонування інформаційної системи підприємства торгівлі.

1.2.2.2 Технологія зовнішніх компонент NativeAPI

Технологія зовнішньої компоненти на основі NativeAPI [22] дозволяє створювати програмні рішення, що будуть динамічно під'єднуватись та тісно взаємодіяти з системою "1С:Підприємство 8", розширюючи її можливості. Зовнішні компоненти дозволяють вирішувати широкий спектр задач, в тому числі, пов'язаних із використанням стороннього торговельного обладнання спільно із програмним комплексом "1С:Підприємство 8". Зовнішні компоненти можуть бути під'єднані як до сервера додатків "1С: Підприємство", так і до клієнтських застосувань, зокрема і до веб-клієнта.

Дана технологія дозволяє створювати зовнішні компоненти, які можуть бути під'єднані як до сервера додатків "1С: Підприємство 8", так і до клієнтських застосувань для операційних систем Windows та Linux, а також мобільних платформ Windows Runtime, Android і iOS [10].

Особливістю реалізації технології для мобільних операційних систем Windows Runtime, Android і iOS є неможливість використання функції інтерфейсу користувача [1].

Проте слід зазначити, що використання механізму зовнішніх компонент NativeAPI є складнішим з точки зору розробки програмного забезпечення, оскільки передбачає реалізацію низькорівневої міжсистемної взаємодії.

1.2.2.3 Механізм COM-з'єднання

Технологічний стандарт COM [2] (Component Object Model — модель компонентного об'єкта) призначений для організації міжпроцесної взаємодії та динамічного створення та обробки об'єктів зовнішніх систем у будь-якій мові програмування, що підтримує дану технологію. Механізм використовується переважно у сімействі операційних систем Microsoft Windows, хоча був реалізований на декількох інших платформах.

Механізм COM-з'єднання забезпечує надійний та швидкий доступ до даних "1С:Підприємство 8" із зовнішніх застосунків, що підтримують дану технологію.

До основних недоліків механізму COM-з'єднання слід віднести:

- складності пов'язані з взаємодією різних версій платформи 1С;
- високі витрати на встановлення з'єднання, при відсутності кешування, що неефективно для великої кількості дрібних викликів;
- прив'язка до сімейства ОС Windows.

Серед переваг даної технології можна виділити:

- відсутність необхідності вносити зміни до конфігурації баз, до яких виконується підключення;
- широкий набір функціональності;
- простіший процес налагодження;
- програмно доступний інтерактивний режим роботи з базою;
- можливість клієнта виступати ініціатором взаємодії з платформою 1С.

1.2.2.4 Засоби OLE Automation серверу

У розробці програмного забезпечення для операційної системи Microsoft Windows технологія OLE Automation є одним з механізмів реалізації міжпроцесного зв'язку [17]. Даний підхід заснований на принципі взаємодії засобами COM-з'єднання, але з метою забезпечення міжсистемної сумісності, технологія OLE Automation обмежується використанням лише підмножини типів COM.

Однак, сумісні з OLE Automation COM-сервери можуть використовувати вбудовану реалізацію серіалізації об'єктів OLE, що дозволяє уникнути потреби створення додаткових проксі-компонентів.

Система "1С: Підприємство" може бути запущена зовнішніми додатками в якості OLE Automation сервера і надає доступ до всіх своїх об'єктах, атрибутів і методів свого глобального контексту. Окрім того, OLE-сервер "1С: Підприємство" має додаткові методи, за допомогою яких можна виконати послідовність операторів або обчислити вираз, заданий на вбудованій мові "1С: Підприємство".

Взаємодія на основі OLE Automation-сервера подібна до взаємодії на основі COM технології. Основні відмінності полягають у наступному:

- у випадку Automation-сервера запускається повноцінний застосунок

"1С:Підприємство 8", а у випадку СОМ-з'єднання – відносно "невеликий" та "легковісний" СОМ-сервер;

- при роботі через СОМ з'єднання недоступні функціональні можливості, пов'язані з інтерфейсом користувача.

1.3 Огляд існуючих програмних рішень

На ринку прикладного програмного забезпечення для інтеграції спеціалізованого торговельного обладнання доступні численні рішення для впровадження окремих типів пристроїв.

Так, наприклад, до найпопулярніших та найпоширеніших рішень для інтеграції мобільних терміналів збору даних належать такі програмні комплекси як "1С-Рарус: Мобільний термінал збору даних v8.3", "Штрих-код Комбайн" та "СКАНКОД.Мобільний Термінал", а для автоматизації робочих місць касирів та продавців засобами впровадження POS-терміналів – система "MyPOS.Каса".

Програмний продукт **"1С-Рарус: Мобільний термінал збору даних v8.3"** виконаний у вигляді додатку, побудованого на мобільній платформі 1С. Ключовими функціональними можливостями додатку є автоматизації бізнес-процесів збору та обробки інформації про фактичну кількість товару у торговельному приміщенні та на складі.

Додаток надає можливість роботи як в онлайн режимі взаємодії з інформаційною базою 1С, так і в оффлайн режимі з подальшим вивантаженням даних.

Суттєвим недоліком даного рішення є неможливість роботи з нетиповими конфігураціями платформи "1С: Підприємство 8", що накладає суттєві обмеження на галузі впровадження.

Програмний комплекс **"Штрих-код Комбайн"** структурно розділений на два ключові компонента: серверний модуль, призначений для функціонування під управлінням операційної системи Windows, та спеціального додатку для пристроїв на базі операційної системи Android. Першочергово програмний комплекс призначений для автоматизації бізнес-процесу інвентаризації та

підтримує роботу як у режимі сканера штрих-кодів, так і безпосередньо в режимі інвентаризації.

Під час роботи у режимі сканера штрих-кодів мобільний пристрій лише передає відскановані значення до серверного модуля, що у свою чергу виконує емуляцію вводу штрих-коду з клавіатури. Режим роботи "інвентаризація" окрім сканування та передачі даних до серверного модуля передбачає відображення інформації про відсканований товар. Проте режим "інвентаризація" вимагає попередньої підготовки даних номенклатури та ручного завантаження результатів роботи до бази даних 1С.

Критичним обмеженням, що "Штрих-код Комбайн" накладає на сферу застосування, є необхідність постійного мережевого з'єднання між серверним модулем та мобільним додатком.

Програмний комплекс для управління складом із залучення терміналів збору даних **"СКАНКОД.Мобільний Термінал"** спрямований на вирішення широкого спектру задач, таких як надходження та відвантаження товару, інвентаризація, переміщення товару тощо. Система складається з додатку для торговельного обладнання під керування операційної системи Android та спеціальної зовнішньої компоненти для взаємодії з платформою 1С.

Така архітектура з одного боку спрощує інтеграцію із типовими конфігураціями платформи, проте з іншого вимагає внесення до них змін, що може негативно вплинути на процес підтримки інформаційного комплексу підприємства в цілому.

Програмний продукт **"MyPOS.Каса"** в першу чергу призначений для автоматизації робочого місця касира засобами впровадження спеціалізованого торговельного обладнання. Проте рішення реалізоване на базі технології WEB-додатків, що дозволяє замінити дорогі спеціалізовані POS-термінали на мобільні пристрої під керуванням операційної системи Android, iOS або Windows з подальшим під'єднанням типового торговельного обладнання (ваг, принтерів етикеток, сканерів штрих-кодів тощо). Система "MyPOS.Каса" може використовуватися як окремий самостійний продукт, так і в комплексі та тісній взаємодії з обліковими системами, такими як "1С: Підприємство 8".

1.4 Постановка задачі інтеграції торговельного обладнання

Змістовний аналіз предметної області в сфері автоматизації бізнес-процесів підприємств торгівлі виявив проблему інтеграції спеціалізованого обладнання до інформаційних систем, побудованих на базі програмної платформи "1С: Підприємство 8", що є одним із лідерів та фактичним монополістом на території країн СНД та України зокрема.

Платформа "1С: Підприємство 8" пропонує вбудовані засоби інтеграції лише типового торговельного обладнання, що належить до переліку офіційно підтримуваного компанією 1С. Проте автоматизації специфічних бізнес-процесів окремих підприємств може вимагати залучення спеціалізованого торговельного обладнання, що не підтримується платформою нативно, такого як мобільні термінали збору даних, сучасні інформаційні дисплеї тощо.

Провівши аналіз існуючих програмних рішень для інтеграції нетипового торговельного обладнання було виявлено, що вони або вимагають внесення змін до існуючої конфігурації платформи 1С, що може негативно впливати на стабільність функціонування інформаційної системи підприємства в цілому, або ж накладають жорсткі обмеження на класи цільових пристроїв та, відповідно, на сфери впровадження.

Саме тому гостро постало питання створення уніфікованого підходу та формалізації механізму інтеграції непідтримуваного нативно платформою "1С: Підприємство 8" торговельного обладнання до інформаційної системи підприємства, що з одного боку не вимагав би внесення змін до функціонуючої конфігурації платформи та дозволив би мінімізувати ризики впровадження, а з іншого боку не накладав би жодних обмежень на цільові класи пристроїв.

Метою дослідження стала розробка такого універсального механізму інтеграції спеціалізованого торговельного обладнання до інформаційної системи підприємства, а також створення прикладної програмної реалізації на його основі.

Об'єктом дослідження є інформаційні системи автоматизації обліку й керування підприємством.

Предметом дослідження є засоби автоматизації складських та торгових процесів шляхом інтеграції спеціалізованого торговельного обладнання до існуючої інформаційної системи підприємства.

Наукова новизна дослідження полягає у створенні універсального механізму міжсистемної взаємодії інформаційної системи на базі платформи "1С: Підприємство 8" та торговельного обладнання, що дозволяє мінімізувати ризики такої інтеграції за рахунок уникнення необхідності внесення змін до поточної конфігурації 1С.

Механізм інтеграції торговельного обладнання із платформою "1С: Підприємство 8" має задовольняти наступним вимогам:

- забезпечення організації тісної взаємодії із об'єктами платформи 1С незалежно від конфігурації;
- функціонування незалежно від платформи "1С:Підприємство 8";
- забезпечення функціонування системи без внесення змін до конфігурації програмного комплексу 1С;
- відсутність обмежень на пристрої класу спеціалізованого торговельного обладнання.

З метою проведення попередньої передексплуатаційної перевірки в дії теоретично обгрунтованого уніфікованого механізму інтеграції спеціалізованого обладнання розробити програмне забезпечення для часткової або повної автоматизації окремих бізнес-процесів товарообігу підприємства торгівлі з використанням терміналів збору даних та дисплеїв покупця, зокрема процесів інвентаризації та роздрібної торгівлі.

Висновки до розділу

В рамках дослідження було проведено аналіз предметної області в сфері автоматизації бізнес-процесів торговельних підприємств, в результаті якого можна зробити наступні висновки:

- проведене дослідження предметної області виявило проблему інтеграції спеціалізованого нетипового торговельного обладнання до інформаційної

системи підприємства на базі платформи "1С: Підприємство 8", що є одним із лідерів серед корпоративних систем автоматизації на території як СНД, так і України зокрема;

- аналіз існуючих програмно-технічних рішень показав, що жодне з них не дозволяє виконати впровадження такого обладнання без втручання у внутрішню структуру інформаційної системи, і як результат – без ризиків порушення ключових процесів підприємства;

- сформульовано задачу розробки уніфікованого механізму інтеграції торговельного обладнання, що дозволив би мінімізувати ризики такого впровадження за рахунок уникнення необхідності внесення змін до існуючої інфраструктури корпоративної системи підприємства торгівлі.

2 МЕТОДИ ТА ЗАСОБИ ПОБУДОВИ УНІФІКОВАНОЇ ІНТЕГРАЦІЙНОЇ ПЛАТФОРМИ

Процесу розробки методів та засобів інтеграції спеціалізованого торговельного обладнання до інформаційної системи підприємства торгівлі передуює аналіз предметної області та умов експлуатації продукту.

Під час інтеграції обладнання до існуючої інфраструктури необхідно провести аналіз поточної інформаційної системи підприємства та розробити механізми інтеграції нового програмного забезпечення, що дозволить мінімізувати ризики впливу на існуючі бізнес-процеси та на стабільність інформаційної системи підприємства в цілому.

2.1 Опис інформаційної системи підприємства

Інформаційна система будь-якого торговельного підприємства, що використовує для автоматизації бізнес-процесів продукт "1С:Підприємство 8", складається з серверу 1С у вигляді як серверного, так і файлового варіанту розгортання системи, та під'єднаних до нього клієнтів (рисунок 2.1).

Сервер 1С відповідальний за забезпечення життєдіяльності інформаційної системи підприємства. Сервер забезпечує централізований доступ до бази даних 1С, виконує задачі обробки та модифікації даних, здійснює контроль та розмежування доступу користувачів до даних. З метою забезпечення відмовостійкості сервер 1С може бути розгорнутим з використанням кластера серверів, а сервер баз даних може бути розміщений на віддалених потужностях. Даний підхід сприяє не лише пришвидшенню роботи системи в цілому за рахунок розподілення та балансування навантаження, а й спрощує процес адміністрування та підтримки програмного комплексу.

У якості клієнтів можуть виступати як і так звані тонкі клієнти 1С, так і різноманітні пристрої, що використовують підтримувані конфігурацією 1С протоколи доступу до даних.

Клієнти під'єднується до сервера 1С з використанням різноманітних, в тому числі і мережевих, протоколів зв'язку, таких як:

- клієнтські додатки 1С, такі як "тонкий" та "товстий" клієнти, мобільний клієнт та веб-клієнт, що забезпечують інтерактивну взаємодію користувача з системою "1С: Підприємство";
- протоколи віддаленого робочого столу RDP (Remote Desktop Protocol) або VNC (Virtual Network Computing), що дозволяють забезпечити віддалену роботу користувача із сервером 1С, використовуючи як настільні робочі станції, так і мобільні пристрої;
- протокол передачі та обміну структурованими повідомленнями в розподілених обчислювальних системах SOAP (Simple Object Access Protocol), що базується на форматі XML, та зазвичай передбачає організацію взаємодії з комплексом 1С засобами WEB-сервісів.
- тощо.

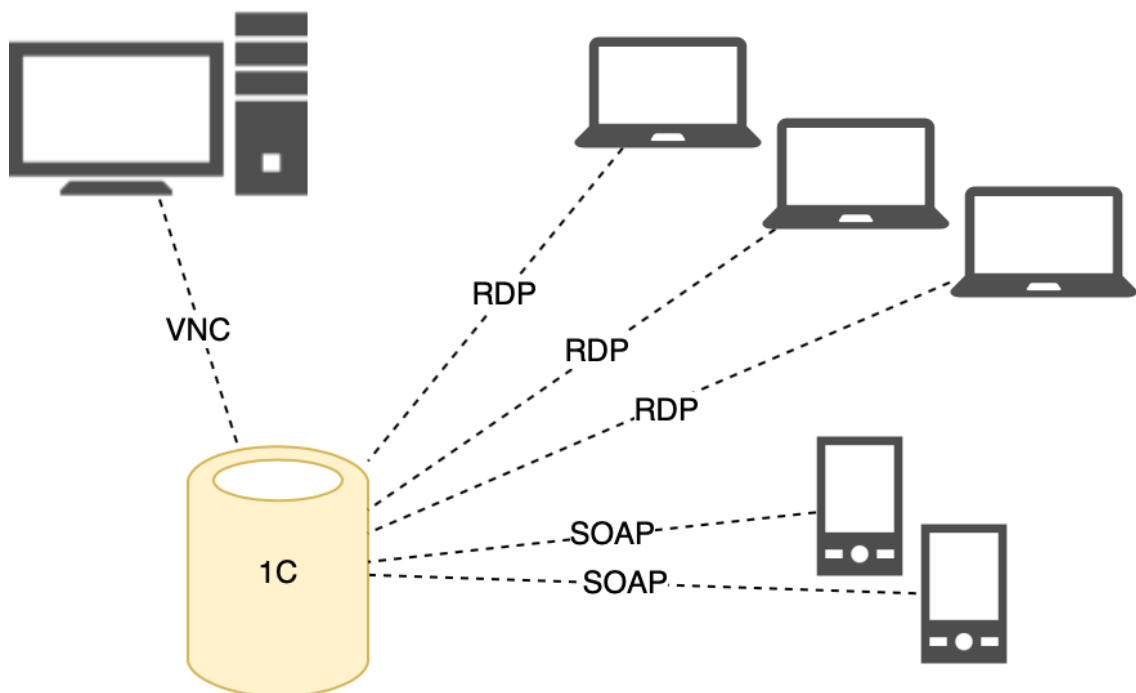


Рисунок 2.1 – Модель типової інформаційної системи підприємства

Проте у випадку необхідності використання спеціалізованого обладнання слід зазначити, що сервер "1С:Підприємство 8" дозволяє використовувати лише обмежений набір торговельного обладнання, що входить до так званого списку

офіційно підтримуваних пристроїв. При використанні стороннього обладнання виникає необхідність внесення змін до існуючої конфігурації, що може призвести до непередбаченої зміни поведінки окремих компонентів та до дестабілізації роботи інформаційної системи в цілому.

2.2 Узагальнений механізм інтеграції торговельного обладнання

Зважаючи на ризики, пов'язані з інтеграцією офіційно непідтримуваного спеціалізованого обладнання до системи на базі платформи "1С: Підприємство 8", доцільним є реалізація необхідного для роботи торговельного обладнання функціоналу за межами існуючої інформаційної мережі підприємства. Така інтеграція передбачає впровадження проміжної ланки між платформою 1С та спеціалізованим обладнанням – так званого сервера автоматизації (рисунок 2.2).

У загальному випадку роль сервера автоматизації полягає в абстрагуванні механізму роботи з даними та безпосередньо зі структурою даних на стороні торговельного обладнання від внутрішньої організації моделі даних з боку платформи 1С. Сервер виконує всі необхідні трансформації та перетворення між моделями даних торговельних пристроїв та конфігурації 1С, беручи на себе відповідальність за доступ та збереження даних на стороні 1С. Сервер автоматизації виступає активною стороною в комунікації з платформою "1С: Підприємство 8", ініціюючи з'єднання та сеанси доступу та обміну даними, що позбавляє конфігурацію 1С необхідності виконувати будь-які дії для підтримки стороннього обладнання, що виходять за межі існуючого функціоналу конфігурації. В такому випадку платформа 1С повністю абстрагована від типів та видів торговельного та іншого спеціалізованого обладнання та лише має надавати уніфікований інтерфейс для обміну даними.

В контексті такої інтеграції доступ до даних інформаційної системи підприємства має забезпечуватись виключно за рахунок підтримуваних платформою 1С протоколів доступу до внутрішньої структури конфігурації, таких як веб-сервіс, COM, тощо.

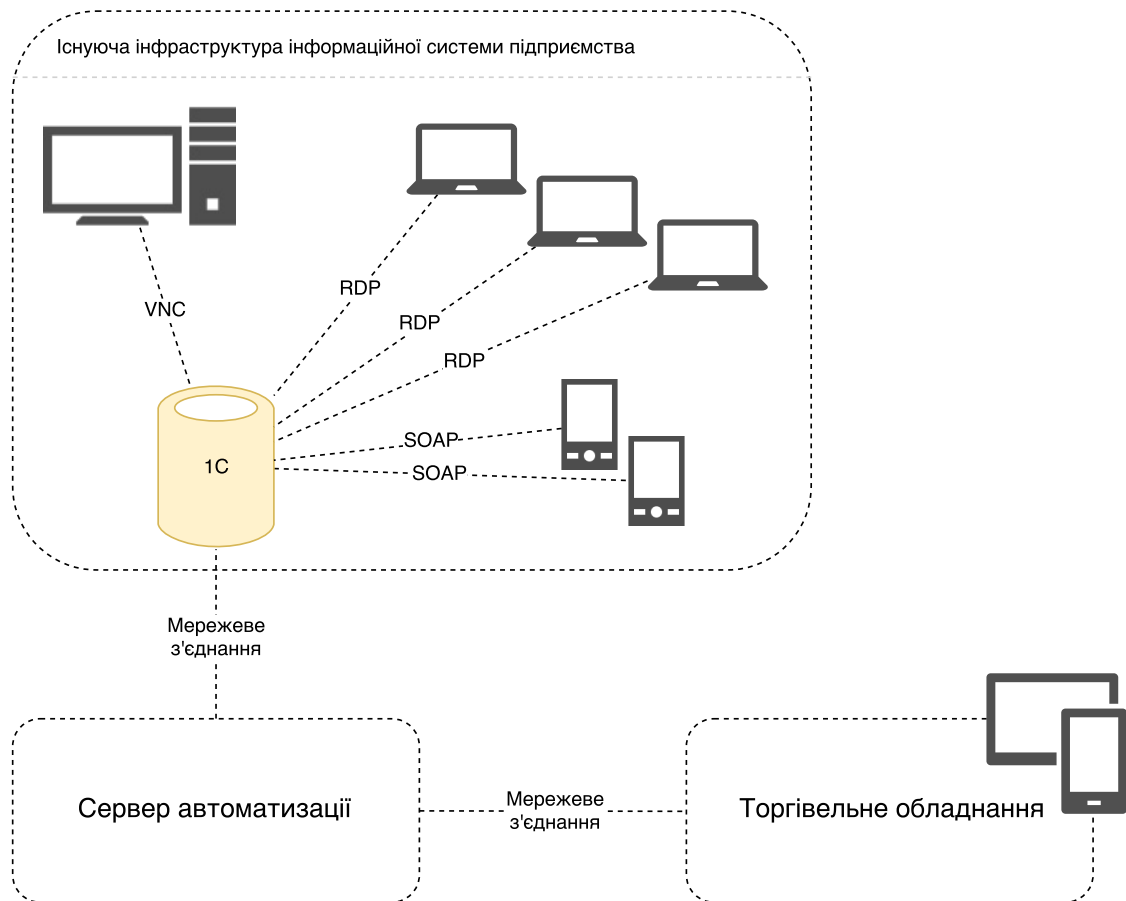


Рисунок 2.2 – Концептуальна модель інтеграції спеціалізованого обладнання

При таких умовах сервер автоматизації виступає посередником між системою "1С:Підприємство 8" та стороннім торгівельним обладнанням, не вимагаючи внесення жодних змін до існуючої конфігурації, тим самим зводячи до мінімуму ризики порушення функціонування існуючих бізнес-процесів підприємства.

Взаємодія серверу автоматизації як з торгівельним обладнанням, так і з платформою 1С відбувається через мережу, як локальну, так і глобальну. Такий підхід дозволяє розмістити сервер автоматизації на відокремлених потужностях, аби мінімізувати ризик негативного впливу як на функціонування серверу 1С зокрема, так і на стабільність інформаційної системи підприємства в цілому.

Сервер автоматизації виконує роль посередника між програмним комплексом "1С: Підприємство" та спеціалізованим обладнанням та забезпечує їх взаємодію без прямої комунікації. Тобто сервер автоматизації забезпечує двосторонній обмін даними між сервером 1С та торгівельним обладнанням, виконуючи за необхідності їх обробку, конвертацію та перетворення,

виступаючи активною стороною взаємодії з платформою 1С.

Даний підхід не накладає жодних обмежень на протоколи міжсистемної взаємодії модулів інформаційного комплексу. Протокол залежить виключно від специфічних потреб підприємства торгівлі, бізнес-процесу, автоматизація якого вимагає залучення додаткового спеціалізованого обладнання та від технічних обмежень, що накладають обрані пристрої.

Впровадження проміжної ланки у вигляді сервера автоматизації суттєво відрізняє дану реалізацію механізму взаємодії торговельного обладнання та інформаційної системи підприємства від існуючих програмно-технічних рішень за рахунок надання ролі активної сторони замість платформи 1С сторонньому програмному забезпеченню, що, в свою чергу, зводить до мінімуму необхідність внесення будь-яких змін до поточної конфігурації програмного комплексу підприємства та повністю абстрагує платформу "1С: Підприємство 8" від стороннього обладнання, а також від моделей даних, що використовуються за межами існуючої інформаційної системи підприємства.

2.3 Концептуальна модель сервера автоматизації

У загальному випадку сервер автоматизації, виступаючи брокером між платформою 1С та спеціалізованим обладнанням, виконується в ізолюваному від 1С контексті, а взаємодія компонентів реалізується лише в рамках мережевої комунікації підсистем програмного комплексу. На абстрактному рівні пристрій встановлює з'єднання з сервером автоматизації, передаючи параметри запиту чи операції. Сервер, в свою чергу, базуючись на власній конфігурації, формує запит до 1С, обробляє результат операції та, виконуючи необхідні перетворення даних, надсилає відповідь на пристрій.

Концептуально сервер автоматизації складається з безпосередньо серверного модуля, засобів його конфігурації, власного локального сховища даних для зберігання поточної конфігурації, параметрів користувачів, тощо, а також уніфікованого компоненту для доступу безпосередньо до даних 1С (рисунок 2.3).

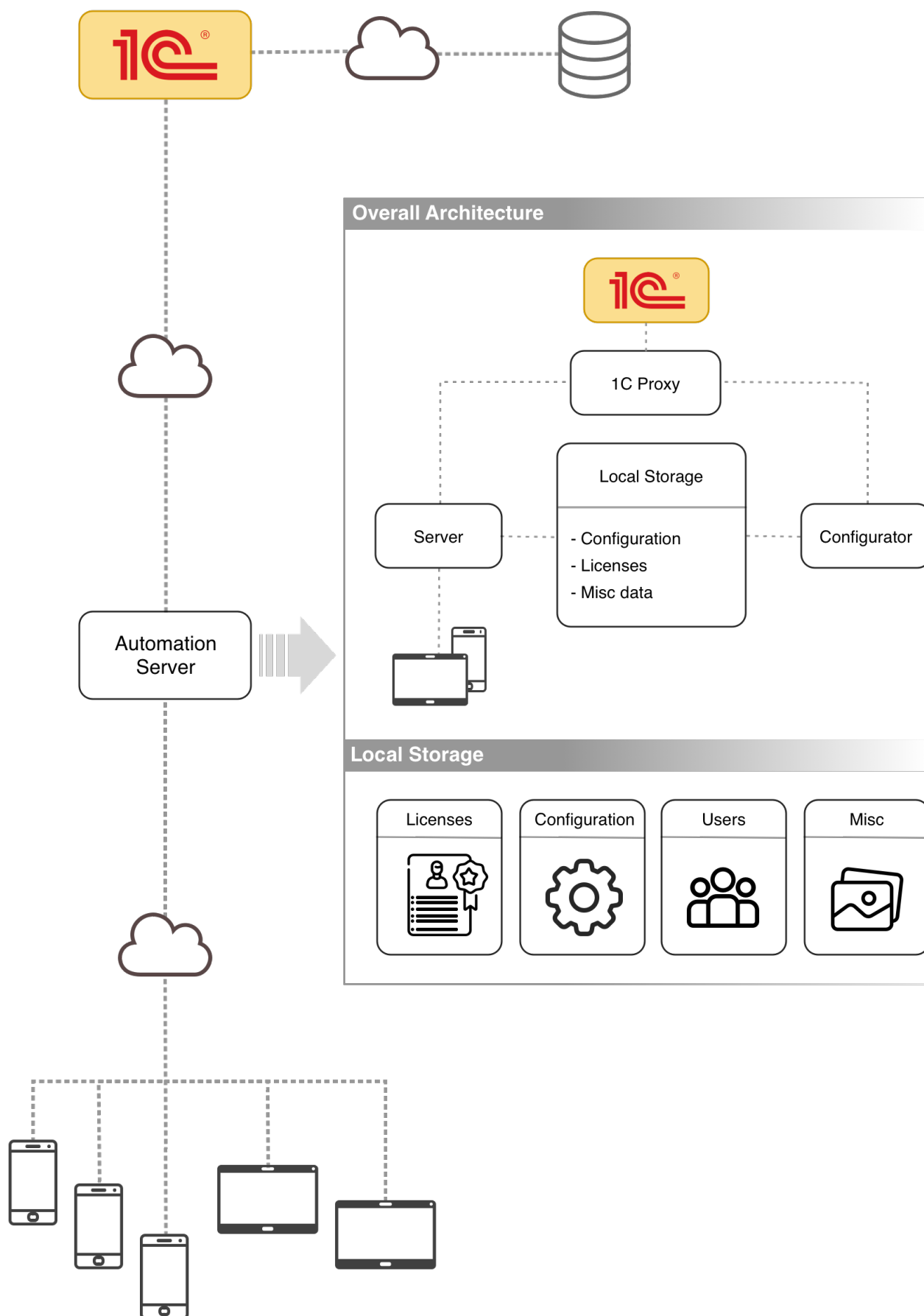


Рисунок 2.3 – Модель сервера автоматизації

Для уніфікованого доступу до інформації програмного комплексу підприємства торгівлі сервер автоматизації будує власну модель даних, створену на базі метаданих, що надаються платформою 1С. За створення моделі даних відповідають засоби конфігурації сервера, представлені у вигляді відокремленого програмного модуля.

Будь-який доступ до інформаційної системи 1С як для отримання метаданих, так і для завантаження вмісту реєстрів, довідників, документів, тощо виконуються виключно засобами так званого проксі – програмного компоненту, що відповідає виключно за комунікацію з 1С.

Сервер автоматизації з однієї сторони надає інтерфейс для комунікації з торгівельним та іншим обладнанням засобами власного проприетарного протоколу, що залежить від особливостей бізнес-процесів підприємства, виконуючи роль пасивної сторони, приймаючи та обробляючи вхідні з'єднання, а з іншої сторони за рахунок компоненту "проксі" виконує доступ до даних платформи 1С, виступаючи ініціатором з'єднання.

Сервер автоматизації має власні, незалежні від платформи 1С, засоби контролю доступу до даних, що базуються на політиках розмежування прав на базі конфігурації користувачів системи. У загальному випадку права доступу включають в себе обмеження на під'єднання до окремих баз даних та конфігурацій 1С, заборону на виконання певних операцій з даними, розмежування користувачів за їх ролями в системі, тощо.

Також сервер автоматизації передбачає наявність механізму захисту від несанкціонованого доступу з неавторизованого обладнання, що базується на ліцензуванні на основі унікальних ідентифікаційних даних кожного окремого пристрою.

Узагальнюючи, сервер автоматизації виступає проміжною ланкою між системою 1С та різноманітним торгівельним обладнанням, надаючи пристроям інтерфейс доступу до даних 1С, при цьому беручи на себе відповідальність за безпосередньо взаємодію в інформаційною системою у якості активної сторони комунікації, не вимагаючи жодних дій з боку програмної системи підприємства. Сервер автоматизації може розміщуватись на відокремлених від існуючої

інформаційної системи потужностях, вимагаючи лише мережевий доступ до програмного забезпечення підприємства. Програмний комплекс має власну конфігурацію, що, поруч з конфігурацією сервера, містить опис структур даних 1С, з якими працює програмний комплекс в рамках автоматизації певного бізнес-процесу підприємства, а також власну систему контролю доступу до даних на базі багатокористувацького режиму, власні засоби моніторингу стану системи, власні механізми захисту від несанкціонованого доступу.

2.3.1 Засоби конфігурації сервера

Сервер автоматизації має власну, відокремлену від платформи 1С, конфігурацію, що зберігається у локальному файлового сховищі та містить відомості про:

- параметри підключення до системи 1С, як до файлового варіанту розгортання, так і до серверного;
- опис структур даних поточної конфігурації 1С, необхідних для автоматизації певного бізнес-процесу;
- правила отримання, обробки та фільтрування даних;
- відомості про користувачів, їх права доступу, політики безпеки;
- відомості про авторизаційні дані пристроїв;
- тощо.

Для налаштування параметрів сервера автоматизації, а також для контролю за поточним станом системи використовується окремий модуль – Конфігуратор, узагальнена будова якого зображена на рисунку 2.4.

Конфігуратор має доступ до локального файлового сховища даних сервера, де містяться власне конфігураційні файли, відомості про користувачів, файли ліцензій, журнали подій.

Доступ до даних платформи "1С: Підприємство" та завантаження метаданих про будову інформаційної бази, необхідних для здійснення конфігураційних дій, забезпечується засобами відокремленого модуля обміну даними – "Проксі 1С".

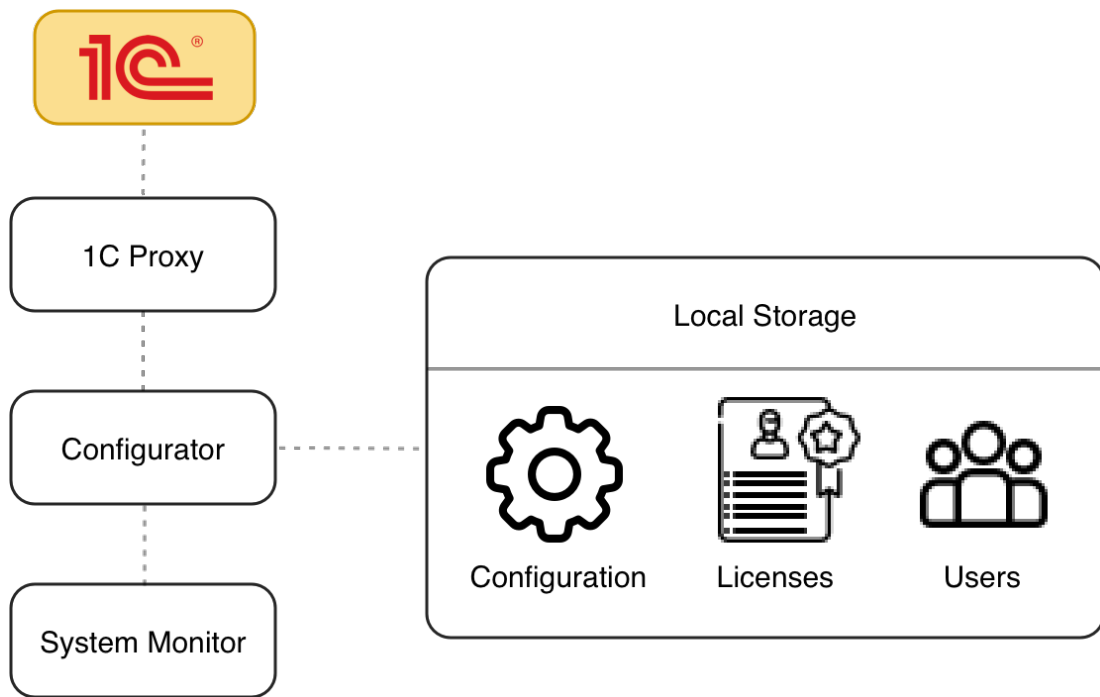


Рисунок 2.4 – Архітектура конфігураційного модуля

Також Конфігуратор передбачає наявність вбудованих модулів управління користувачами, керування правами доступу до даних та моніторингу стану системи.

Конфігуратор не має засобів прямої взаємодії з сервером автоматизації. Єдиний механізм комунікації – редагування та оновлення конфігураційних файлів сервера, що, в свою чергу, призведе до відповідних нотифікацій зі сторони серверного модуля та реакції сервера на ці нотифікації – оновлення правил роботи відповідно до завантаженої конфігурації.

2.3.2 Багатокористувацький режим роботи

Сервер автоматизації передбачає власний, незалежний від 1С, контроль доступу до даних на основі багатокористувацького режиму. Відомості про користувачів входять до базової конфігурації системи та включають в себе:

- загальні відомості про користувача: ім'я, ідентифікатор, тощо;
- опції авторизації в системі;
- політики контролю доступу до даних: обмеження на виконання певних операцій та доступу до окремих баз даних 1С.

Архітектура передбачає підтримку механізму авторизації, абстрагованого

від конкретного способу ідентифікації користувача на пристрої. Спосіб авторизації може залежати від можливостей обладнання, з якого виконується доступ до системи (наприклад, пара логін-пароль, 2D або 3D штрих-код, тощо) та не впливає на уніфікований спосіб ідентифікації користувача системи.

2.3.3 Міжмережева клієнт-серверна взаємодія

Взаємодія компонентів системи побудована на клієнт-серверній архітектурі та відбувається виключно засоби мережевого з'єднання, що не накладає жодних обмежень на розміщення елементів інформаційної системи в рамках розгортання програмного комплексу.

В контексті сеансу взаємодії з платформою "1С: Підприємство 8" сервер автоматизації виступає клієнтом та ініціює процес комунікації використовуючи лише підтримувані системою 1С протоколи.

В рамках взаємодії сервера із спеціалізованим торговельним обладнанням пристрої виступають клієнтами та є ініціаторами з'єднання. Комунікація відбувається за рахунок проприетарних протоколів зв'язку, специфічних для кожного окремого бізнес-процесу.

Висновки до розділу

В результаті дослідження методів та засобів побудови уніфікованої платформи для інтеграції спеціалізованого торговельного обладнання до інформаційної системи підприємства можна зробити наступні висновки:

- проведено аналіз типової інформаційної системи торговельного підприємства на базі програмного комплексу "1С: Підприємство 8" та обґрунтовано доцільність реалізації платформи для інтеграції стороннього обладнання за межами існуючої інформаційної мережі;
- спроектовано концептуальну модель уніфікованого механізму побудови міжсистемної взаємодії платформи "1С: Підприємство 8" та спеціалізованого обладнання (рисунок 2.3), що базується на впровадженні проміжної ланки у вигляді сервера автоматизації.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРИКЛАДНОГО РІШЕННЯ

Проектуванню та розробці програмного комплексу для вирішення задач автоматизації бізнес-процесів підприємства торгівлі за рахунок інтеграції спеціалізованого обладнання передують моделювання цільових бізнес-процесів, аналіз потоків даних та визначення учасників і їх ролей.

3.1 Структурне моделювання програмного забезпечення

Впровадження проміжної ланки між платформою 1С та інформаційною системою підприємства торгівлі у вигляді сервера автоматизації передбачає появу окремих технічних процесів з метою конфігурації, адміністрування та підтримки інтегрованої системи.

3.1.1 Адміністрування інтеграційної системи

Адміністрування сервера автоматизації включає в себе наступні групи задач (рисунок 3.1):

- керування сервером;
- керування користувачами та пристроями;
- конфігурація окремих модулів автоматизації.

Керування сервером автоматизації передбачає можливість ручного запуску чи зупинки сервера, налаштування параметрів автоматичного запуску, моніторинг загального стану системи в певний момент часу, створення звітів про несправності, тощо.

Адміністрування сервера включає в себе конфігурацію користувачів системи, їх ролей та прав доступу, параметрів і дозволених способів авторизації. Налаштування ролей користувачів передбачає надання окремим обліковим записам доступ лише до певних можливостей системи з метою розмежування зон відповідальності.

Управління пристроями передбачає можливість контролю доступу до

системи лише окремим пристроям або групам пристроїв на базі системи ліцензування. Адміністратору надається можливість управління файлами ліцензій: реєстрація нових та видалення існуючих.

В рамках управління пристроями також можливий контроль за поточними сесіями роботи з системою, що включає в себе як моніторинг поточних підключень, так і можливість ручного завершення сеансів роботи.

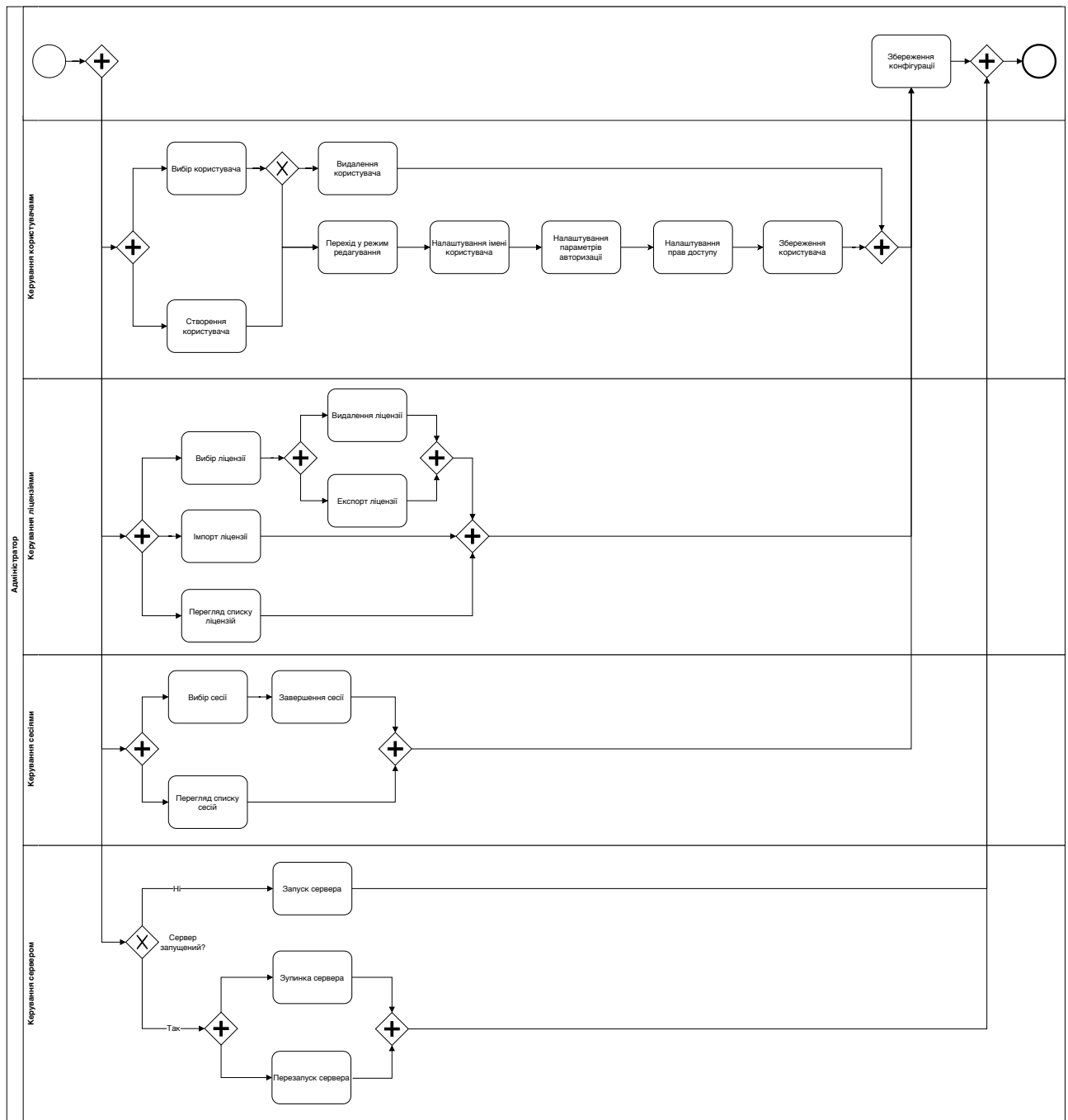


Рисунок 3.1 – Структурна схема процесу адміністрування сервера автоматизації

3.1.2 Процеси товарообігу в контексті роздрібної торгівлі

Товарообіг в контексті функціонування підприємства роздрібною торгівлі є складним та комплексним процесом, що включає в себе як підмножину задач внутрішнього обігу та обліку товару в межах підприємства, так і безпосередньо групу процесів роздрібною торгівлі.

3.1.2.1 Процес інвентаризації

Одним із ключових та критичних етапів внутрішнього товарообігу підприємства роздрібною торгівлі є процес інвентаризації, що ставить на меті забезпечення цілісності та актуальності даних щодо наповнення цільовим товаром як окремих підрозділів, так і підприємства в цілому.

В контексті сервера автоматизації бізнес-процес інвентаризації товару структурно складається з двох незалежних етапів:

- конфігурація сервера автоматизації;
- власне проведення інвентаризації.

Конфігурація модуля інвентаризації сервера автоматизації виконується адміністратором інформаційної системи і здебільшого полягає в налаштуванні профілів інвентаризації (рисунок 3.2).

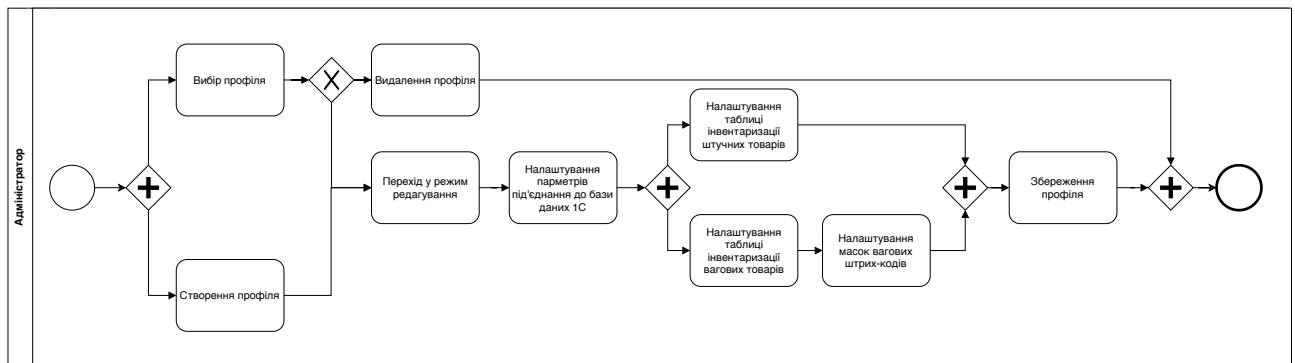


Рисунок 3.2 – Структурна схема процесу конфігурації параметрів інвентаризації

Профіль інвентаризації – це логічна сутність системи, що об'єднує параметри доступу до бази даних, правила роботи з даними в рамках поточного підключення, а також містить інші, необхідні для інвентаризаційного процесу, опції.

Керування профілями інвентаризації на високому рівні зводиться до трьох операцій:

- створення та налаштування нового профіля;
- редагування існуючого профіля;
- видалення профіля.

Процес проведення інвентаризації може бути розпочатий лише після успішного конфігурування сервера автоматизації. Першим та єдиним конфігураційним етапом є налаштування параметрів підключення до сервера. Після вдалого під'єднання та авторизації виконавця інвентаризації в системі відбувається завантаження доступних документів з урахуванням поточних опцій фільтрування та прав доступу (рисунок 3.3).

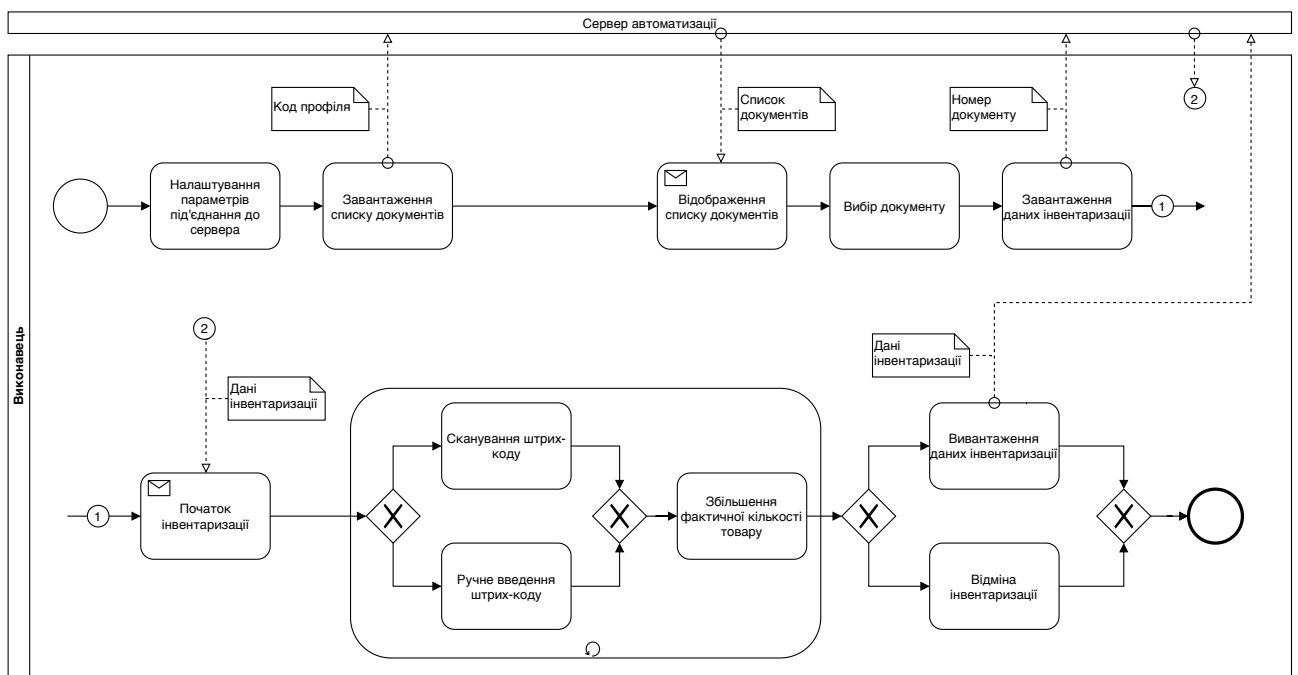


Рисунок 3.3 – Структурна схема бізнес-процесу проведення інвентаризації

Після вибору виконавцем необхідного документу інвентаризації відбувається завантаження даних на пристрій. Проведення інвентаризації відбувається шляхом послідовного сканування штрих-кодів товарів. В окремих випадках можливе ручне введення штрих-кодів. У відповідності від типу штрих-коду відбувається одна з наступних дій:

- у разі штучного штрих-коду відбувається збільшення кількості цільового товару на одиницю;
- у разі вагового товару збільшення кількості відбувається на значення, вказане у штрих-коді.

При завершенні інвентаризації відбувається запис всіх оброблених інвентаризаційних даних до 1С з урахуванням усіх налаштованих пост-обробок даних.

Виконавець у будь-який момент має можливість перервати проведення інвентаризації із втратою всіх поточних інвентаризаційних даних.

3.1.2.2 Дисплей покупця

На ряду з інвентаризацією одним з важливих етапів товарообігу в контексті роздрібної торгівлі є взаємодія з кінцевим споживачем в рамках реалізації цільового товару. Обов'язковою умовою ведення роздрібної торгівлі в багатьох країнах СНД, зокрема в Україні, є наявність інформаційного табло з даними про поточний товар, що опрацьовується касиром – так званого дисплея покупця.

Використання системи дисплея покупця передбачає наявність трьох етапів (рисунок 3.4):

- конфігурація пристроїв;
- налаштування промо-матеріалів;
- інтеграція з платформою "1С: Підприємство 8.

Конфігурація пристроїв передбачає налаштування з боку сервера автоматизації груп дисплеїв, їх прив'язка до окремих касових місць, тощо.

Головна задача дисплея покупця – це відображення інформації про поточний чек. Для більш ефективного використання пристроїв в моменти простою дисплей має можливість відображати так звані рекламні промо-матеріали. Для налаштування цієї можливості адміністратору інформаційної системи необхідно завантажити медіа-матеріали у підтримуваному форматі, налаштувати порядок та параметри їх відтворення, тощо. Після завантаження медіа-файлів на цільові пристрої відтворення промо-матеріалів почнеться автоматично.

Інтеграція з платформою "1С: Підприємство 8" полягає в налаштуванні надсилання спеціальних сигналів з боку 1С до сервера автоматизації з інформацією про зміни стану поточного чеку засобами емуляції програмного принтеру штрих-кодів без суттєвих змін конфігурації системи.

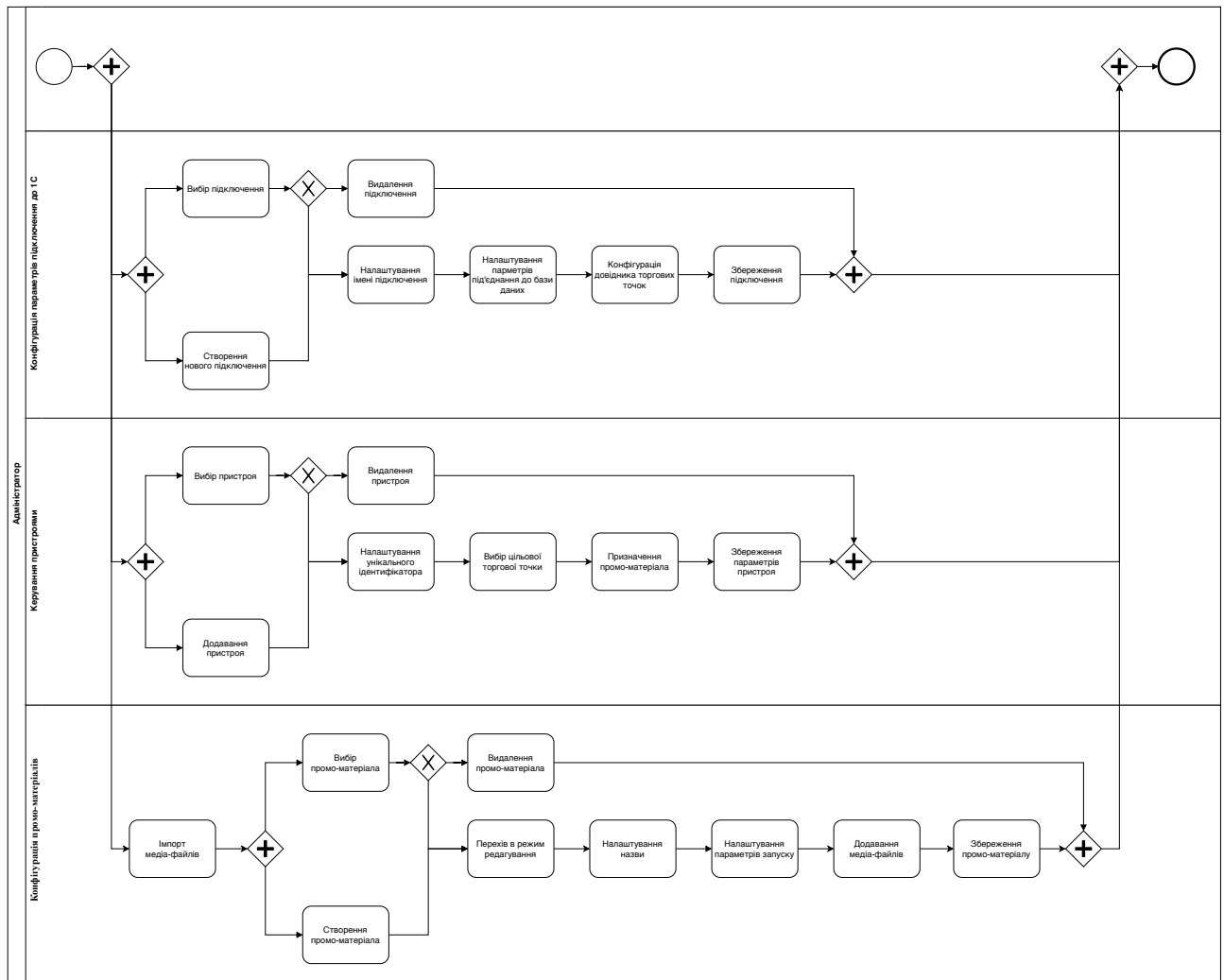


Рисунок 3.4 – Структурна схема процесу конфігурації параметрів дисплея покупця

3.2 Опис програмної реалізації

Базуючись на розробленому універсальному механізмі інтеграції спеціалізованого торговельного обладнання до інформаційної системи підприємства було створено програмний комплекс для часткової або повної автоматизації окремих бізнес-процесів товарообігу, зокрема інвентаризації та роздрібної торгівлі. Програмний комплекс передбачає наявність трьох основних складових: сервера автоматизації включно з додатком для здійснення адміністративних та конфігураційних дій, групи мобільних застосунків для торговельного обладнання та веб-серверу, призначеного для здійснення віддаленого логіювання та генерації ліцензій (рисунок 3.5).

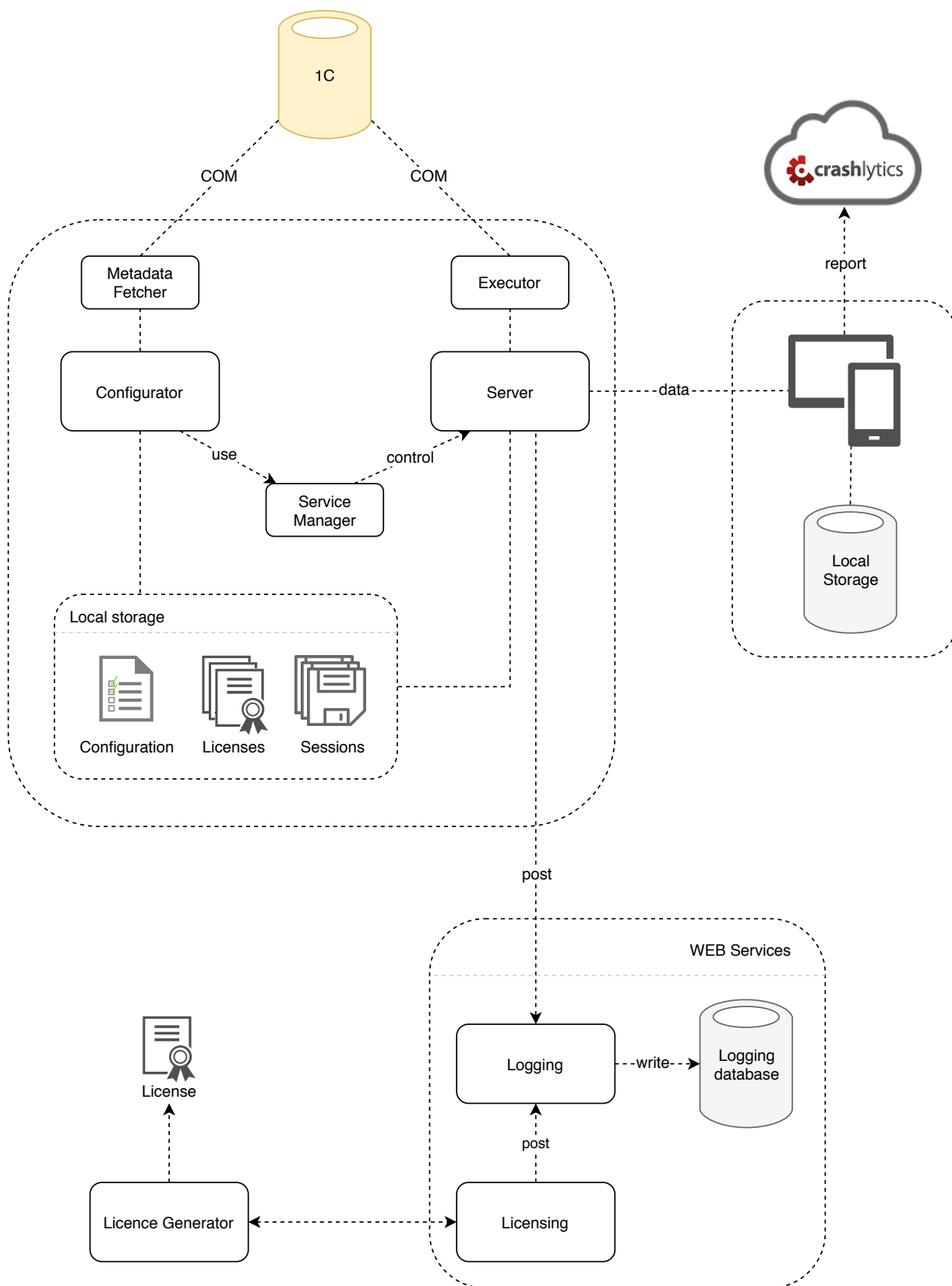


Рисунок 3.5 – Архітектура програмного забезпечення

На абстрактному рівні сервер автоматизації складається з двох основних структурних елементів: власне сервера та засобів його адміністрування і конфігурування.

Сервер автоматизації представлений у вигляді служби Windows.

Взаємодія сервера з платформою "1С: Підприємство 8" реалізована через уніфікований механізм, представлений у вигляді окремого модуля (див. рисунок 3.5 – "Executor"). Взаємодія з торговельним обладнанням базується на технології віддаленого виклику процедур Apache Thrift.

У якості конфігураційних та адміністративних засобів виступає окремий додаток – "Конфігуратор". Конфігуратор дозволяє керувати станом сервера автоматизації, здійснювати контроль за загальним станом системи, поточними сеансами роботи з сервером, налаштовувати параметри окремих елементів системи, правила роботи з платформою "1С: Підприємство 8", тощо. Для налаштування параметрів та правил роботи з даними системи 1С Конфігуратор використовує метадані, для отримання яких призначений окремий модуль – "Metadata fetcher".

Також система передбачає засоби контролю доступу до даних на базі системи ліцензування. Ліцензія дозволяє обмежити доступ до сервера автоматизації, надавши можливість встановлення з'єднання лише обмежених групі пристроїв. Кожна ліцензія однозначно ідентифікує лише один пристрій: використовуючи його унікальний ідентифікатор – серійний номер. Для запобігання зміни серійних номерів пристроїв задля повторного використання ліценцій передбачено механізм сесій. Кожна сесія відображає окремий сеанс роботи з сервером автоматизації і унеможливорює одночасне використання декількох пристроїв з однаковими ідентифікаторами.

Для безпечної генерації ліцензій використовується окремий застосунок, що є лише клієнтом, призначеним для віддаленої генерації та завантаження ліцензій з сервера.

3.2.1 Сервісно-орієнтована будова програмного рішення

Інтеграція обладнання до інформаційної системи підприємства торгівлі передбачає взаємодію з різними за призначенням та методами комунікації і передачі даних класами пристроїв, що мають пряму кореляцію із групами та типами задач і бізнес-процесів підприємства. Саме тому доцільним є

архітектурне розділення внутрішніх компонентів сервера автоматизації на логічні групи – так звані "сервіси" – кожна з яких з однієї сторони відповідає за взаємодію з окремим класом пристроїв, а з іншої – за автоматизацію певного процесу підприємства.

Архітектурно сервер автоматизації складається з певного набору відокремлених компонентів – сервісів – та інфраструктурних елементів, що забезпечують їх повноцінне функціонування (рисунк 3.6).

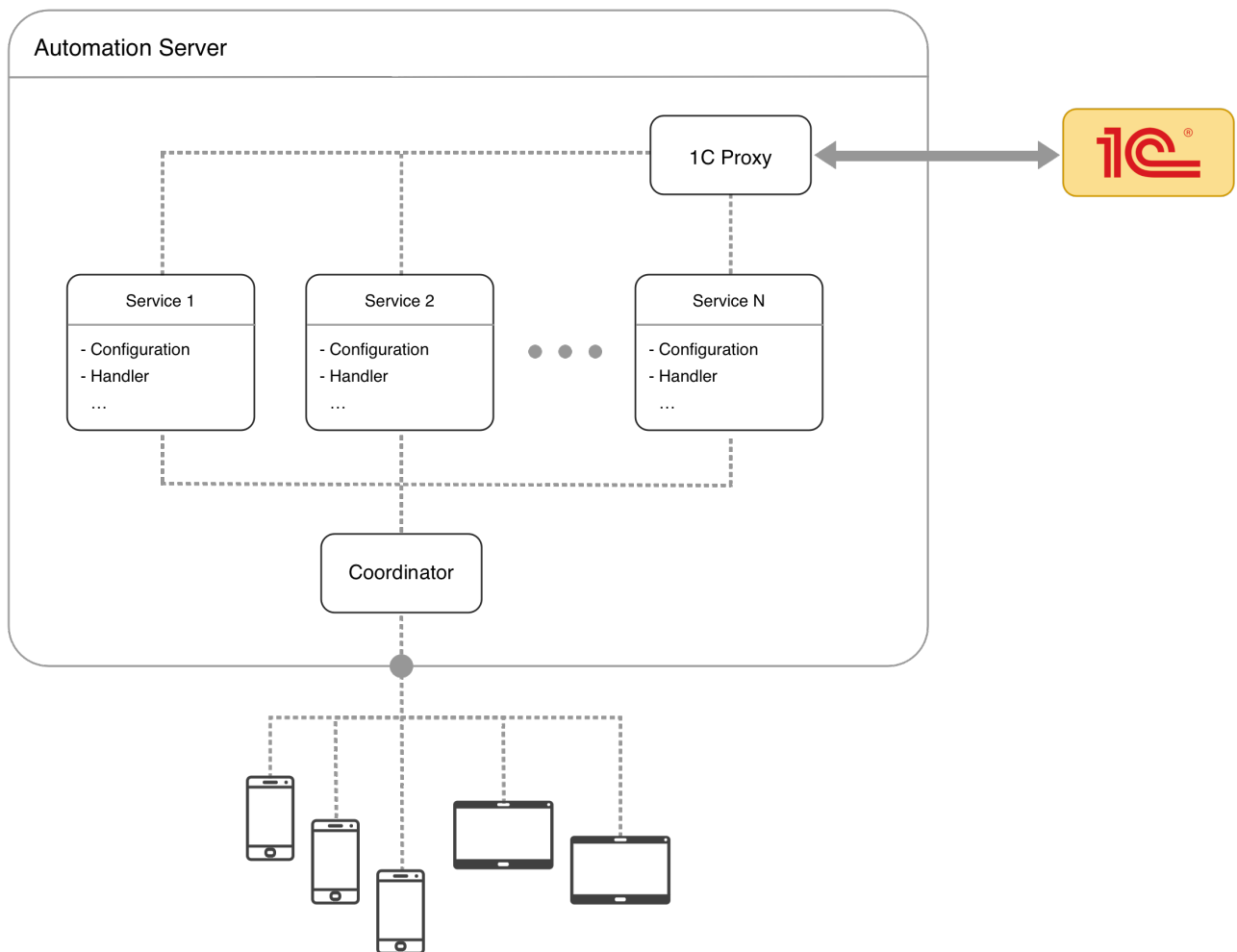


Рисунок 3.6 – Сервісно-орієнтована архітектура програмного забезпечення

У загальному випадку кожен сервіс має власну конфігурацію, власну точку входу запитів на виконання тих чи інших дій, власний ізольований контекст виконання. На абстрактному рівні основана задача сервісу полягає в опрацюванні запиту від пристрою, виконанні всіх необхідних дій з платформою "1С: Підприємство 8" та поверненні результату виконання операції.

Всі вхідні з'єднання опрацьовуються компонентом "Coordinator", що

визначає сервіс, якому на опрацювання буде передано запит. "Coordinator" також відповідає за такі безпекові елементи системи, як встановлення безпечного мережевого з'єднання, аутентифікація пристрою та користувача, тощо.

Сервіс відповідає лише за логіку та модель поведінки сервера, пов'язану з автоматизацією певного бізнес-процесу підприємства торгівлі, та не несе жодної відповідальності за будь-які інші дії з боку сервера автоматизації, що виходять за межі цільового бізнес-процесу.

Сервер автоматизації не передбачає механізмів міжсервісної взаємодії, надаючи кожному сервісу незалежний ізольований контекст виконання, забезпечуючи кожен сервіс необхідними системними ресурсами.

Модульний підхід до організації внутрішньої структурної сервера автоматизації забезпечує можливість безпечного розширення можливостей системи з метою автоматизації нових бізнес-процесів підприємства торгівлі шляхом інтеграції нових сервісів без впливу на існуючі компоненти системи, а також збільшує стабільність програмного рішення в цілому за рахунок ізоляції контекстів виконання обробників різних сервісів.

3.2.1.1 Сервіс "Інвентаризація"

Структурно сервіс складається з двох основних компонентів: серверного модуля інвентаризації та мобільного додатку. Основною задачею серверного модуля є організація взаємодії між кінцевим користувачем (мобільним додатком) та платформою "1С: Підприємство 8" згідно правил, заданих конфігурацією сервісу. Мобільний додаток призначений для безпосередньо проведення інвентаризації – тобто для сканування штрих-кодів цільових товарів з метою оцінки фактичної кількості залишків.

Серверний модуль включає в себе наступні компоненти:

- підмодуль конфігурування з власною моделлю даних конфігурації;
- підмодуль роботи з базою даних 1С;
- підмодуль міжмережевої взаємодії, призначений для забезпечення комунікації з терміналами збору даних задля передачі даних інвентаризації;
- власне компонент інвентаризації, що відповідає за логіку проведення процесу інвентаризації згідно з поточною конфігурацією сервісу.

3.2.1.2 Сервіс "Дисплей покупця"

Структурно сервіс складається з двох основних компонентів: серверного модуля дисплея покупця та мобільного додатку. Основною задачею серверного модуля є опрацювання подій 1С та інформування мобільного клієнта про оновлення даних поточного чеку, а також надання клієнту необхідних для відображення медіа-матеріалів та правил роботи з ними. Мобільний додаток, у свою чергу, відповідає за формування та відображення у реальному часі інформації про поточний чек на базі даних, наданих серверним модулем, а під час простою – за відображення промо-матеріалів.

Серверний модуль включає в себе наступні компоненти:

- підмодуль конфігурування з власною моделлю даних конфігурації;
- підмодуль роботи з базою даних 1С;
- підмодуль міжмережевої взаємодії, призначений для забезпечення комунікації з дисплеєм покупця для актуалізації даних щодо поточного чеку та для передачі графічних та відео-файлів.

Серверний модуль сервісу "Дисплей покупця" не має власної логіки роботи чи механізмів опрацювання даних, а є лише проміжною ланкою між системою "1С: Підприємство 8" та спеціалізованим обладнанням, абстрагуючи платформу 1С від особливостей комунікації з різними типами пристроїв.

3.2.2 Клієнтські мобільні застосунки

До складу кожного сервісу входить відповідний мобільний застосунок, що працює лише у тісній взаємодії із сервером автоматизації. Кожен клієнт призначений для роботи лише з одним сервісом.

3.2.2.1. Мобільний застосунок "Інвентаризація"

Мобільний додаток сервісу "Інвентаризація" розроблений у вигляді застосунку для торгівельного обладнання на базі операційної системи Android та є ключовою і невід'ємною частиною програмного комплексу автоматизації бізнес-процесів підприємства торгівлі. Додаток не є незалежним програмним елементом – його використання можливе лише в умовах тісної взаємодії із сервером автоматизації.

Основним завданням та головним призначенням мобільного додатку є завантаження, модифікація та збереження даних інвентаризації у відповідності до поточної конфігурації, наданої сервером автоматизації, що задовольняє вимоги щодо автоматизації окремого бізнес-процесу обліку цільового товару підприємства торгівлі.

На абстрактному рівні основним процесом як інвентаризації, так і функціонування додатку, є підтримка актуальності даних щодо фактичної кількості цільового товару у торговельному приміщенні чи складі. З цією метою застосунок надає засоби сканування штрих-кодів товарів. Спосіб передачі та опрацювання штрих-кодів побудований на базі механізму Android Broadcast Intents, що дозволяє змінювати провайдера штрих-кодів в залежності від типу чи класу пристрою без необхідності модифікації вихідного коду застосунку, абстрагуючи програмний рівень від апаратного. Такий підхід дозволяє портувати програмне забезпечення на будь-який пристрій з вбудованим сканером штрих-кодів з мінімізацією ресурсів, необхідних для адаптації застосунку під нове апаратне забезпечення.

Для збереження стану інвентаризації (тобто фактичних даних щодо кількості товарних одиниць у торговельному приміщенні) використовується база даних на основі СКБД SQLite [11], схема якої зображена на рисунку 3.7.

Таблиця `NomenclatureItems` призначена для збереження кількісних відомостей про дані інвентаризації. Для кожної номенклатурної одиниці зберігаються наступні характеристики:

- облікова кількість (`actual_amount`);
- фактична кількість на момент початку інвентаризації (`planned_amount`);
- кількість відсканованого за поточну сесію товару (`amount_delta`);
- позначка, що вказує, чи є номенклатурна одиниця ваговим товаром;
- список значень реквізитів номенклатури.

У таблиці `Barcodes` зберігаються завантажені з бази даних 1С штрих-коди штучних товарів та коди вагових товарів (`barcode`). Ця таблиця встановлює відповідність між номенклатурною одиницею таблиці інвентаризації (`item_id`) та штрих-кодом (чи кодом вагового товару). Кожен елемент таблиці

NomenclatureItems може мати будь-яку кількість відповідних йому штрих-кодів.

У таблиці AttributesDescriptors зберігається опис реквізитів елементів таблиці інвентаризації. Кожен дескриптор містить назву (name) та позначку (is_primary), що встановлює пріоритет в опрацюванні та відображенні певного атрибуту. Параметр target дозволяє зберігати різні групи дескрипторів атрибутів, встановлюючи приналежність запису в таблиці до певної категорії дескрипторів. Значення параметру name_index дозволяє керувати порядком опрацювання та відображення дескрипторів.

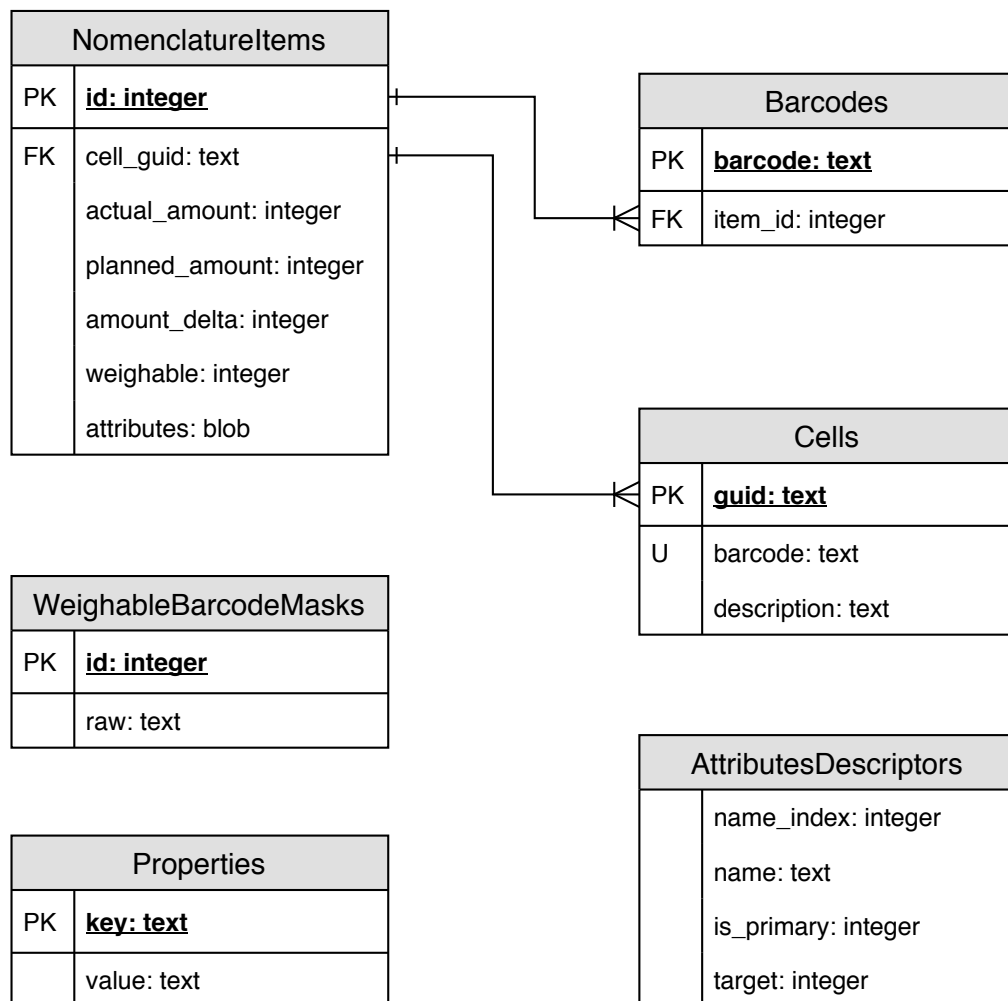


Рисунок 3.7 – Схема бази даних мобільного застосунку "Інвентаризація"

Таблиця Properties призначена для зберігання додаткових параметрів проведення інформації у форматі словника ключ – значення (key – value). Так, наприклад, таблиця містить відомості про поточну сесію інвентаризації, її ідентифікатор, параметри опрацювання дробових чисел, опції їх округлення, тощо.

3.2.2.2 Мобільний застосунок "Дисплей покупця"

Мобільний застосунок сервісу "Дисплей покупця" розроблений для роботи на планшетних пристроях на базі операційної системи Android. Його основним завданням є відображення актуальної інформації про поточний чек у режимі реального часу. Формат відображення задається конфігурацією сервісу та передається сервером автоматизації на кожен пристрій окремо. У моменти простою кожен дисплей відображає сконфігуровані промо-матеріали.

Для забезпечення роботи у режимі реального часу сервер автоматизації тримає постійне мережеве з'єднання з кожним пристроєм. Механізм мережевої взаємодії передбачає регулярні спеціальні повідомлення для швидкого реагування на порушення стабільності каналу зв'язку та переініціалізацію з'єднання.

Формат відображення даних включає в себе структуру інформаційної сітки даних, шрифти, кольори, тощо. Формат повністю контролюється конфігурацією сервісу з боку сервера автоматизації та може бути оновлений у будь-який момент часу з відповідним оновленням на всіх пристроях у реальному часі.

Для збереження промо-матеріалів мобільний застосунок має власне локальне сховище даних та базу даних на основі СКБД SQLite [12] для систематизованого збереження інформації про медіа-файли.

3.2.3. Міжмережева взаємодія компонентів системи

Міжмережева комунікація компонентів система побудована на основі технології віддаленого виклику процедур Apache Thrift [1], що дозволяє працювати як через локальну мережу, так і через мережу Internet.

3.2.3.1 Взаємодія із терміналами збору даних

З метою забезпечення каналу зв'язку між спеціалізованим торгівельним обладнанням та сервером автоматизації в рамках автоматизації бізнес-процесу інвентаризації було розроблено мережевий сервіс на базі протоколу Apache Thrift (таблиця 3.1), що надає набір методів для двосторонньої передачі інвентаризаційних даних.

Метод `GetInventoryDocuments` слугує для отримання списку доступних для проведення інвентаризації документів з бази даних платформи "1С: Підприємство 8". Метод передбачає встановлення лімітів на максимальний обсяг даних (тобто, кількості документів), що будуть передані сервером. Повертає масив документів інвентаризації.

Метод `FetchInventoryData` призначений для власне завантаження даних інвентаризації – масиву номенклатурних одиниць з їх обліковою кількісною характеристикою. Повертає масив інвентаризаційних даних, що включають:

- таблицю інвентаризації;
- довідних штрих-кодів та кодів вагових товарів;
- облікову кількісну характеристику номенклатури;
- фактичну кількісну характеристику номенклатури.

Виклик цього методу у разі відсутності помилок його опрацювання призводить до створення сесії інвентаризації на стороні сервера автоматизації. Ідентифікатор створеної сесії передається мобільному клієнту та повинен бути використаний для успішного завершення сеансу інвентаризації.

Основною задачею методу `PushInventoryData` є вивантаження таблиці інвентаризації на сервер для її подальшого опрацювання та запису до бази даних платформи 1С.

Для відміни поточної сесії з втратою всіх інвентаризаційних даних призначений метод `DismissInventorySession`.

Виклик методів `PushInventoryData` та `DismissInventorySession` призводить до закриття сесії інвентаризації з боку сервера автоматизації.

Метод `DestroyInventorySession` використовується для примусового завершення сеансу інвентаризації у разі, якщо виникли помилки при опрацюванні відповіді сервера в процесі завантаження інвентаризаційних даних. У такому разі ідентифікатор сесії невідомий клієнту і виклик методу `DismissInventorySession` неможливий.

Головним призначенням методу `TestConnection` є перевірка правильності налаштування параметрів під'єднання до сервера та можливості встановлення з'єднання з сервером автоматизації.

Таблиця 3.1 – Структура сервісу InventoryService

| | | |
|----------------|-------------------------------|------------------|
| Метод | GetInventoryDocuments | |
| Тип результату | GetInventoryDocumentsResponse | |
| Параметри | Назва | Тип |
| | token | ConnectionToken |
| | profileCode | string |
| | notOlderThan | i64 |
| | limit | i32 |
| | timeout | i32 |
| Метод | FetchInventoryData | |
| Тип результату | FetchInventoryDataResponse | |
| Параметри | Назва | Тип |
| | token | ConnectionToken |
| | session | InventorySession |
| | timeout | i32 |
| Параметри | nomenclature | Nomenclature |
| | timeout | i32 |
| Метод | PushInventoryData | |
| Тип результату | PushInventoryDataResponse | |
| Параметри | Назва | Тип |
| | token | ConnectionToken |
| | session | InventorySession |
| | nomenclature | Nomenclature |
| | timeout | i32 |
| Метод | DismissInventorySession | |
| Тип результату | ResponseStatus | |
| Параметри | Назва | Тип |
| | token | ConnectionToken |
| | session | InventorySession |
| | timeout | i32 |

Продовження таблиці 3.1

| | | |
|----------------|-------------------------|-----------------|
| Метод | DestroyInventorySession | |
| Тип результату | ResponseStatus | |
| Параметри | Назва | Тип |
| | token | ConnectionToken |
| | timeout | i32 |
| Метод | TestConnection | |
| Тип результату | ResponseStatus | |
| Параметри | Назва | Тип |
| | token | ConnectionToken |
| | profileCode | string |
| | timeout | i32 |

Усі методи мережевого сервісу InventoryService, призначеного для роботи з клієнтськими застосунками на терміналах збору даних в рамках автоматизації бізнес-процесу інвентаризації, у якості вхідних параметрів приймають параметр token типу ConnectionToken та timeout.

Параметр token містить дані про ініціатора мережевого з'єднання, тобто:

- ідентифікаційну інформацію про пристрій, з якого ініціюється з'єднання;
- ідентифікатор сесії інвентаризації у разі, якщо сесія була створена;
- версію протоколу обміну даними для гарантування правильності обробки переданих даних;
- відмітку часу;
- тощо.

Також параметр token виступає безпековим елементом з'єднання, так як містить додаткові відомості про ключі мережевої взаємодії.

Числовий параметр timeout необхідний для обмеження максимального часу опрацювання клієнтського запиту сервером.

Типи даних, що використовуються в рамках мережевої взаємодії торговельного обладнання та сервера автоматизації, описані у таблицях 3.2 – 3.9.

Об'єкт ConnectionToken містить параметри, необхідні для встановлення безпечного мережевого з'єднання, такі як ідентифікатор цільового сервісу, версію API клієнта, ідентифікатор пристрою, ключ з'єднання, відмітку часу тощо (таблиця 3.2).

Таблиця 3.2 – Структура об'єкта ConnectionToken

| Назва атрибуту | Тип | Обов'язковий |
|----------------|--------|--------------|
| key | binary | Так |
| serviceName | string | Так |
| apiVersion | string | Так |
| deviceId | string | Так |
| timestamp | i64 | Так |
| sessionId | string | Ні |

Структура даних ResponseStatus містить дані загальні для всіх типів запитів дані відповіді сервера автоматизації, такі як статус виконання операції, опис помилки у разі неуспішного опрацювання клієнтського запиту, ідентифікатор сесії (таблиця 3.3).

Таблиця 3.3 – Структура об'єкта ResponseStatus

| Назва атрибуту | Тип | Обов'язковий |
|----------------|------------|--------------|
| code | StatusCode | Так |
| message | string | Ні |
| sessionId | string | Ні |

Структура даних Attribute описує будь-яке значення, отримане з бази даних 1С та складається з двох полів – коду об'єкту та значення (таблиця 3.4).

Таблиця 3.4 – Структура об'єкта Attribute

| Назва атрибуту | Тип | Обов'язковий |
|----------------|--------|--------------|
| code | string | Ні |
| value | string | Так |

Об'єкт типу `AttributeDescriptor` призначений для збереження та передачі опису способу відображення даних у клієнтському застосунку. Містить назву атрибуту та його характеристики (таблиця 3.5).

Таблиця 3.5 – Структура об'єкта `AttributeDescriptor`

| Назва атрибуту | Тип | Обов'язковий |
|----------------|--------|--------------|
| name | string | Так |
| primary | bool | Так |

Основним призначенням структури `NomenclatureItem` є збереження даних та параметрів номенклатурної одиниці. Кожен об'єкт типу `NomenclatureItem` містить список відповідних йому штрих-кодів чи кодів вагових товарів, облікову та фактичну кількості, список значень реквізитів та опис інших характеристик (таблиця 3.6).

Таблиця 3.6 – Структура об'єкта `NomenclatureItem`

| Назва атрибуту | Тип | Обов'язковий |
|----------------|-----------------|--------------|
| barcodes | set<string> | Так |
| actualAmount | i64 | Так |
| plannedAmount | i64 | Так |
| amountDelta | i64 | Так |
| weighable | bool | Так |
| attributes | list<Attribute> | Так |

Тип даних `InventoryDocument` призначений для передачі відомостей про документ інвентаризації, що включають код документу, дату його створення, список інформаційних реквізитів документу тощо (таблиця 3.7).

Таблиця 3.7 – Структура об'єкта `InventoryDocument`

| Назва атрибуту | Тип | Обов'язковий |
|----------------|-----------------|--------------|
| code | string | Так |
| date | i64 | Так |
| attributes | list<Attribute> | Так |

Тип даних `InventorySession` призначений для опису поточної сесії взаємодії з сервером автоматизації в рамках сервісу інвентаризації та містить дані про поточний профіль і документ (таблиця 3.8).

Таблиця 3.8 – Структура об'єкта `InventorySession`

| Назва атрибуту | Тип | Обов'язковий |
|--------------------------------|--------------------------------|--------------|
| <code>profile</code> | <code>string</code> | Так |
| <code>inventoryDocument</code> | <code>InventoryDocument</code> | Так |

Для збереження параметрів роботи системи використовується в цілому використовується структура даних `Options` (таблиця 3.9). Так, наприклад, значення параметру `decimalPlaces` вказує на кількість знаків після коми, що будуть використовуватись системою для дробових обчислень, а опція `weighableBarcodeMasks` призначена для передачі списку масок вагових товарів.

Таблиця 3.9 – Структура об'єкта `Options`

| Назва атрибуту | Тип | Обов'язковий |
|------------------------------------|---------------------------------|--------------|
| <code>decimalPlaces</code> | <code>i8</code> | Так |
| <code>weighableBarcodeMasks</code> | <code>list<string></code> | Ні |

Таблиці 3.10 – 3.12 містять опис структур даних, що використовуються як контейнери для значень, що повертаються відповідними методами сервісу інвентаризації. Кожна структура містить відомості про виконання операції типу `ResponseStatus`, а також, у разі успішного виконання операції, специфічні для кожного окремого запиту дані.

Таблиця 3.10 – Структура об'єкта `GetInventoryDocumentsResponse`

| Назва атрибуту | Тип | Обов'язковий |
|------------------------------------|--|--------------|
| <code>status</code> | <code>ResponseStatus</code> | Так |
| <code>documents</code> | <code>list<InventoryDocument></code> | Ні |
| <code>attributesDescriptors</code> | <code>list<AttributeDescriptor></code> | Ні |

Таблиця 3.11 – Структура об'єкта FetchInventoryDataResponse

| Назва атрибуту | Тип | Обов'язковий |
|--------------------------------|---------------------------|--------------|
| status | ResponseStatus | Так |
| nomenclature | Nomenclature | Ні |
| countableAttributesDescriptors | list<AttributeDescriptor> | Ні |
| weighableAttributesDescriptors | list<AttributeDescriptor> | Ні |
| options | Options | Ні |

Таблиця 3.12 – Структура об'єкта PushInventoryDataResponse

| Назва атрибуту | Тип | Обов'язковий |
|----------------|----------------|--------------|
| status | ResponseStatus | Так |
| failedItems | Nomenclature | Ні |

3.2.3.1 Взаємодія з дисплеями покупця

Мережева комунікація сервера автоматизації із дисплеями покупця складається з двох елементів:

- постійне мережеве з'єднання, що забезпечує неперервну роботу в режимі реального часу (рисунок 3.8);
- періодичне оновлення відомостей про конфігурації промо-матеріалів та власне завантаження за потребою медіа-файлів.

Першим етапом встановлення мережевого з'єднання є надсилання клієнтом запиту Init, передаючи параметри пристрою, безпекові ключі з'єднання. У разі правильної конфігурації параметрів під'єднання до сервера клієнт отримує відповідь InitResponse, що містить або підтвердження сервером правильності параметрів, або ж відмову та її причину. У разі успішного початкового обміну повідомленнями з'єднання вважається встановленим. В іншому разі клієнт повторює спробу.

Після встановлення з'єднання сервер може керувати станом клієнта, надсилаючи йому повідомлення SetContent та Clear, тим самим передаючи дані про поточний чек. На кожен запит клієнт надсилає відповідь, що містить статус опрацювання операції.

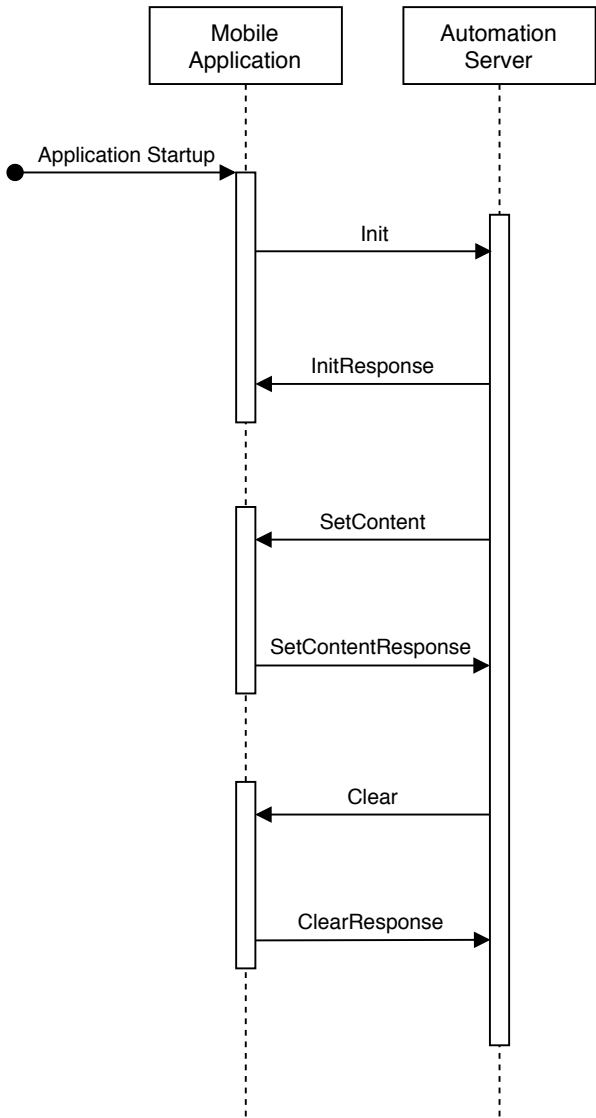


Рисунок 3.8 – Схема мережевої взаємодії із дисплеєм покупця

Періодично клієнт надсилає запит для оновлення даних про промо-матеріали та параметри їх відображення. У разі зміни параметрів з боку сервера клієнту надсилаються нові дані. Для опису порядку відтворення матеріалів, режиму анімування тощо використовуються структури даних, описані у таблицях 3.13 – 3.17.

Таблиця 3.13 – Структура сервісу MediaService

| | | |
|----------------|--------------------------|-----------------|
| Метод | GetMediaSequence | |
| Тип результату | GetMediaSequenceResponse | |
| Параметри | Назва | Тип |
| | token | ConnectionToken |

Таблиця 3.14 – Структура об'єкта GetMediaSequenceResponse

| Назва атрибуту | Тип | Обов'язковий |
|----------------|----------------|--------------|
| status | ResponseStatus | Так |
| sequence | MediaSequence | Ні |

Таблиця 3.15 – Структура об'єкта MediaSequence

| Назва атрибуту | Тип | Обов'язковий |
|------------------|-------------------|--------------|
| objects | list<MediaObject> | Так |
| placeholder | MediaObject | Ні |
| continuationMode | ContinuationMode | Так |
| startDelay | i64 | Так |

Таблиця 3.16 – Структура об'єкта MediaObject

| Назва атрибуту | Тип | Обов'язковий |
|----------------|-----------|--------------|
| name | string | Так |
| filename | string | Так |
| type | MediaType | Так |
| animation | Animation | Так |
| duration | i64 | Ні |

Таблиця 3.17 – Структура об'єкта Animation

| Назва атрибуту | Тип | Обов'язковий |
|----------------|---------------|--------------|
| type | AnimationType | Так |
| duration | i64 | Так |

3.2.3.3 Механізм завантаження метаданих

В рамках налаштування сервісів, підтримуваних сервером автоматизації, виникає необхідність отримання метаданих з описом структур даних поточної конфігурації платформи "1С: Підприємство 8".

Для отримання метаданих використовується окремий сервіс на базі технології Apache Thrift – MetadataFetcher, методи якого описані в таблиці 3.18.

Таблиця 3.18 – Структура сервісу MetadataFetcher

| | | |
|----------------|--|------------|
| Метод | TestConnection | |
| Тип результату | void | |
| Параметри | Назва | Тип |
| | connection | Connection |
| Призначення | Перевірки правильності параметрів під'єднання. | |
| Метод | GetDocuments | |
| Тип результату | list<string> | |
| Параметри | Назва | Тип |
| | connection | Connection |
| Опис | Повертає перелік доступних документів. | |
| Метод | GetDocumentContentColumns | |
| Тип результату | list<string> | |
| Параметри | Назва | Тип |
| | connection | Connection |
| | document | string |
| Призначення | Отримання переліку табличних частин. | |
| Метод | GetDataRegisters | |
| Тип результату | list<string> | |
| Параметри | Назва | Тип |
| | connection | Connection |
| Призначення | Завантаження списку регістрів відомостей. | |
| Метод | GetDocumentAttributes | |
| Тип результату | list<Attribute> | |
| Параметри | Назва | Тип |
| | connection | Connection |
| | document | string |
| | limit | i8 |
| Призначення | Завантаження переліку реквізитів документу. | |

Продовження таблиці 3.18

| | | |
|----------------|---|------------|
| Метод | GetDocumentContentAttributes | |
| Тип результату | list<Attribute> | |
| Параметри | Назва | Тип |
| | connection | Connection |
| | document | string |
| | content | string |
| | limit | i8 |
| Призначення | Отримання реквізитів табличної частини документу. | |
| Метод | GetDataRegisterAttributes | |
| Тип результату | list<Attribute> | |
| Параметри | Назва | Тип |
| | connection | Connection |
| | dataRegister | string |
| | limit | i8 |
| Призначення | Завантаження реквізитів реєстру відомостей. | |

Типи даних, що використовуються у процесі мережевої взаємодії між конфігуратором та відокремленим компонентом Metadata Fetcher, описані у таблицях 3.19 та 3.20.

Таблиця 3.19 – Структура об'єкта Connection

| Назва атрибуту | Тип | Обов'язковий |
|------------------|-------------------|--------------|
| version | ConnectionVersion | Так |
| connectionString | string | Так |

Таблиця 3.20 – Структура об'єкта Attribute

| Назва атрибуту | Тип | Обов'язковий |
|----------------|-----------------|--------------|
| name | string | Так |
| children | list<Attribute> | Ні |

3.2.4 Архітектура WEB-серверу

Одним із компонентів сервера автоматизації є WEB-сервер, до головних завдань якого входять:

- віддалене логіювання та ведення журналу подій в рамках мережевої взаємодії елементів системи;
- віддалена генерація ліцензій.

Для збереження журналу подій використовується база даних на основі СКБД MySQL [7], що містить єдину таблицю Events, структура якої описана у таблиці 3.21.

Таблиця 3.21 – Структура таблиці Events

| Поле | Тип |
|---------|--------------|
| id | int |
| date | datetime |
| source | varchar(64) |
| event | varchar(64) |
| origin | varchar(64) |
| content | varchar(256) |

Кожен запис містить унікальний ідентифікатор, дані про дату та час логіювання події (date), джерело надходження даних (сервер, клієнт, сервіс генерації ліцензій тощо), тип (event) та опис (content) події, природу її виникнення (origin). Тип події – це назва сервісу, який ініціював логіювання. Джерело події – це мережеві ідентифікаційні дані пристрою, що виконує віддалене логіювання – ір адреса сервера автоматизації чи пристрою, на якому виконується клієнтський застосунок.

До подій, що заносяться до бази даних, належать усі запити між клієнтськими застосунками та сервером автоматизації, а також результат їх виконання, події генерації ліцензій, різноманітні помилки роботи системи.

Генерація ліцензій реалізована на базі протоколу HTTPS [13] з обмеженням та контролем доступу до сервісу на базі авторизаційних даних у вигляді пари логін – пароль.

3.3 Аналіз безпеки даних

З метою обмеження та контролю доступу до даних розроблено механізм ідентифікації торговельного обладнання на базі ліцензування.

Ліцензії поширюються у вигляді шифрованих бінарних файлів з цифровим підписом. Шифрування виконується із застосуванням симетричного блочного алгоритму шифрування AES у режимі CBC з розміром блоку 128 біт [6]. Цифровий підпис виконується асиметричним криптографічним алгоритмом RSA [16].

У таблиці 3.22 наведено детальний опис елементів ліцензійного файлу.

Таблиця 3.22 – Структура ліцензії

| Поле | Тип | Опис |
|--------------|------------|--|
| version | short | Версія ліцензії |
| id | byte-array | Унікальний ідентифікатор ліцензії, 160-бітове хеш-значення (sha1) від рядка, що базується на значеннях полів device та service |
| token | byte-array | Ключ, sha256 значення від рядка, що базується на значеннях полів device та service |
| service | string | Назва сервісу, доступ для якого надає ліцензія |
| device | string | Ідентифікатор пристрою (серійний номер) |
| creationDate | date | Дата генерації ліцензії |
| owner | string | Назва організації, що згенерувала ліцензію |
| comment | string | Необов'язковий опис ліцензії |

Для запобігання повторному використанню ліцензій для декількох мобільних пристроїв засобами підміни серійного номеру розроблено механізм сесій, що дозволяє обмежити одночасне під'єднання декількох клієнтів з однаковими ідентифікаційними даними.

З метою обмеження доступу до внутрішньої структури програмного комплексу (деасемблювання виконуваних файлів) було проведено обфускацію окремих модулів системи [14].

Висновки до розділу

В результаті моделювання та конструювання програмного комплексу для автоматизації окремих бізнес-процесів підприємства торгівлі можна зробити наступні висновки:

- проаналізовано ключові бізнес-процеси товарообігу підприємства торгівлі та проведено їх декомпозицію з метою систематизації побудови прикладного програмного рішення;

- змодельовано та розроблено сервісно-орієнтовану клієнт-серверну архітектуру прикладного програмного рішення для автоматизації бізнес-процесів інвентаризації та роздрібної торгівлі із залученням терміналів збору даних та дисплеїв покупця на основі спроектованого уніфікованого механізму інтеграції;

- на основі спроектованої архітектури розроблено прикладне програмне забезпечення, що включає в себе наступні компоненти: сервер автоматизації, конфігураційний додаток, клієнтські мобільні застосунки та web-сервер для віддаленого логіювання;

- проаналізовано безпекову складову розробленого програмного рішення та побудовано механізм контролю доступу до даних на основі ліцензування.

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Завершальним етапом розробки будь-якого програмного продукту та, зокрема, системи автоматизації складських та торгових процесів, є його інтеграція до існуючої інфраструктури та інформаційної системи підприємства, а також подальший супровід програмного комплексу з метою усунення недоліків, виявлених в процесі експлуатації, а також задля вдосконалення програмного продукту у відповідності до особливостей його використання в конкретних умовах ринку.

4.1 Розгортання програмного забезпечення

Програмне забезпечення для автоматизації складських та торгових процесів складається з двох основних компонентів:

- безпосередньо сервера автоматизації, включно з додатком для здійснення адміністративних та конфігураційних дій;
- мобільних додатків для торгівельного обладнання, що поставляються у вигляді інсталяційного пакету *.msi [5] та файлів Android додатків *.apk [21] відповідно.

Схема розгорнутого програмного комплексу зображена на рисунку 4.1.

Сервер автоматизації та конфігуратор спроектовані для роботи в середовищі .Net 4.5 [19] (та новіше) під управлінням операційної системи Windows 7 або новішої. перед початком роботи з сервером автоматизації необхідно встановити платформу "1С:Підприємство 8" необхідної версії. Для встановлення сервера автоматизації необхідно запустити виконуваний файл "ASAP Server Installer.msi" та слідувати підказкам інсталятора. В результаті буде встановлена системна служба Windows [15], застосунок для виконання конфігураційних та адміністративних дій "Конфігуратор", а також будуть створені відповідні ярлики.

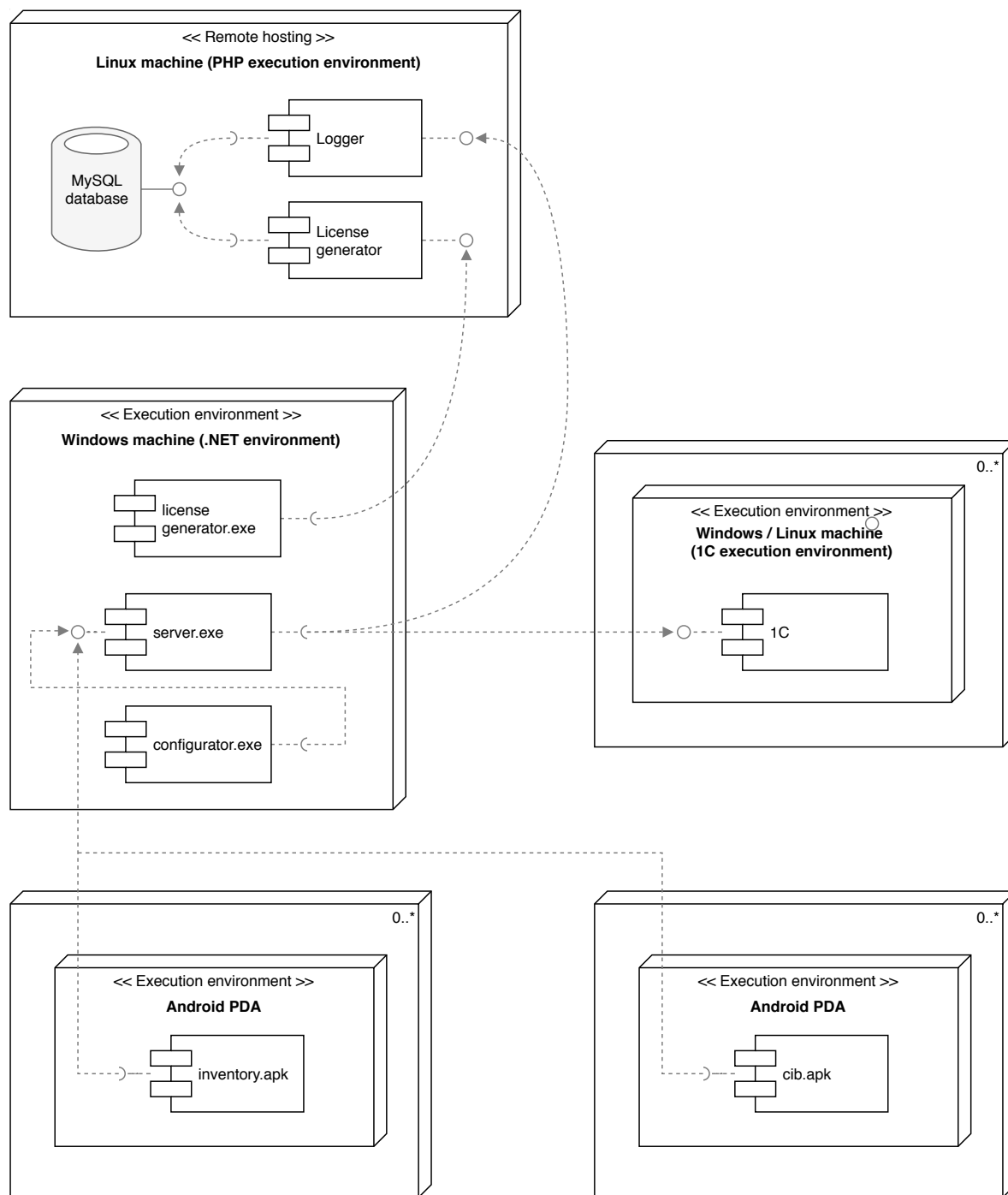


Рисунок 4.1 – Схема структурна розгортання програмного комплексу

Мобільні додатки розроблені для функціонування на спеціалізованому торговельному обладнанні під управлінням операційної системи Android версії 4.0 або новішої. Для початку роботи із застосунками їх необхідно встановити на пристрій будь-яким доступним для платформи Android чином (наприклад, завантаживши додаток з сайту розробника або завантаживши його на пристрій через USB з'єднання).

Для коректної роботи програмного комплексу необхідне мережеве з'єднання для взаємодії із сервером логіювання та генерації ліцензій. Сервер не входить до комплекту поставки програмного забезпечення, залишаючись під контролем та розгорнутим на потужностях розробника.

Платформа "1С:Підприємство 8" може буде розміщена як на одній машині із сервером автоматизації, так і на різних машинах за умови наявності локального мережевого з'єднання або можливості доступу до бази даних 1С через мережу Internet.

4.2 Експлуатація програмного забезпечення

Програмне забезпечення для автоматизації складських та торгових процесів призначене для інтеграції стороннього торгівельного обладнання до існуючої інфраструктури підприємства без ризиків порушення функціонування ключових бізнес-процесів.

Система складається з двох основних компонентів:

- сервера автоматизації, що включає у себе безпосередньо сервер, який встановлюється як служба Windows, та окремий Windows-додаток для здійснення адміністративних та конфігураційних дій;
- мобільних застосунків для торгівельного обладнання на базі операційної системи Android.

4.2.1 Керування сервером автоматизації

Конфігурація та управління сервером автоматизації виконується за допомогою відповідного додатку – "Конфігуратора". Для виконання окремих дій можуть знадобитися права адміністратора.

4.2.1.1 Встановлення сервера

Для встановлення сервера необхідно запустити інсталяційний пакет "ASAP Server Installer (x86).msi" або "ASAP Server Installer (x64).msi» в залежності від необхідної розрядності і виконувати підказки інсталятора.

В результаті буде встановлена системна служба та конфігураційний

додаток, а також будуть створені відповідні ярлики на робочому столі та в папці "ASAP Solutions ▶ Сервер ASAP" в меню Пуск.

4.2.1.2 Конфігурація сервера

Налаштування параметрів сервера автоматизації виконується в програмі "Конфігуратор", ярлик на яку знаходиться в папці "ASAP Solutions ▶ Сервер ASAP" в меню Пуск або на Робочому столі.

Керування користувачами

Деякі сервіси підтримують роботу у багатокористувацькому режимі. Для конфігурації користувачів необхідно в основних параметрах перейти на вкладку "Управління користувачами" (рисунок 4.2).

Для створення нового користувача слід натиснути "+" і вказати ім'я користувача, його пароль, чи дозволено авторизація за допомогою сканування унікального особистого штрих-коду, а також, за необхідності, встановити додатковий авторизаційний штрих-код. У правій частині вікна необхідно налаштувати права доступу користувача до сервісів.

У випадку "Інвентаризації" (див. розділ 4.2.2) необхідно вказати доступні користувачеві профілі інвентаризації, а також встановити порядок їх слідування. Сконфігурований порядок списку профілів встановлює порядок відображення елементів керування на ТСД при авторизації відповідним користувачем. Зміна порядку виконується за допомогою перетягування елементів списку. Також можлива зміна імені та коментаря профілю для кожного конкретного користувача. Для цього в режимі конфігурації користувача необхідно виконати подвійне натискання на назві профілю або коментарі в таблиці "Управління профілями інвентаризації" та вказати нове значення, яке буде використано при формуванні списку профілів на ТСД при авторизації конкретним користувачем.

Також передбачена можливість авторизації без користувача ("Користувач за замовчуванням"), що не залежить від типу ліцензії. Користувач за замовчуванням також підтримує авторизацію по штрих-коду, але пароль не може бути використаний.

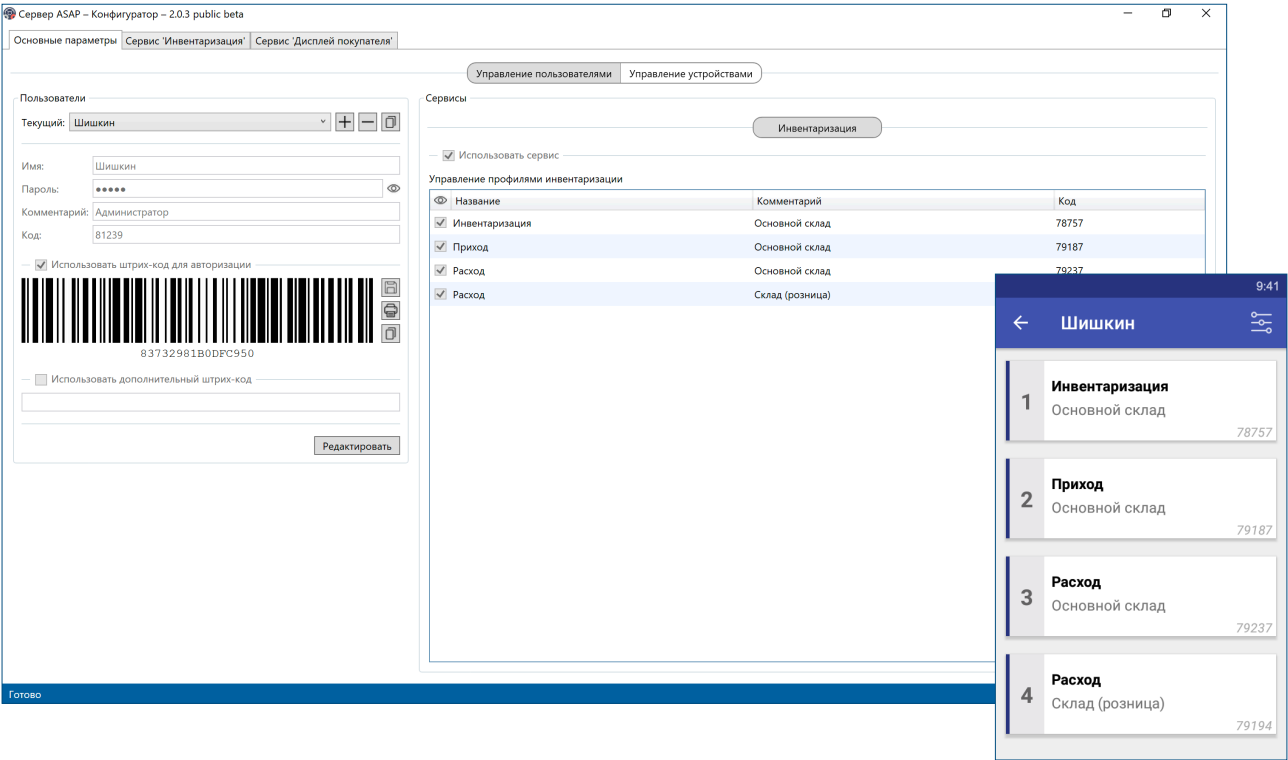


Рисунок 4.2 – Управління користувачами

Керування пристроями

Вкладка "Управління пристроями" (рисунок 4.3) надає інструменти для управління ліцензіями і поточними сесіями роботи з сервером.

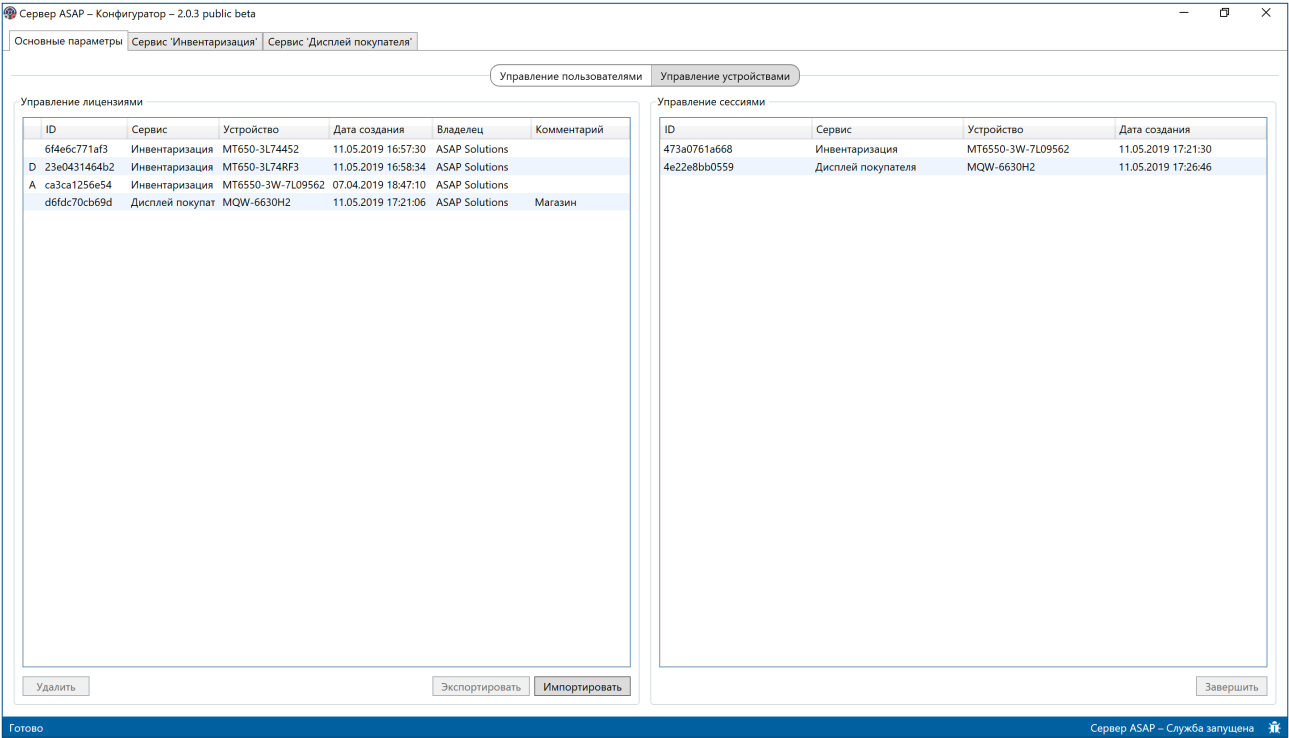


Рисунок 4.3 – Управління пристроями

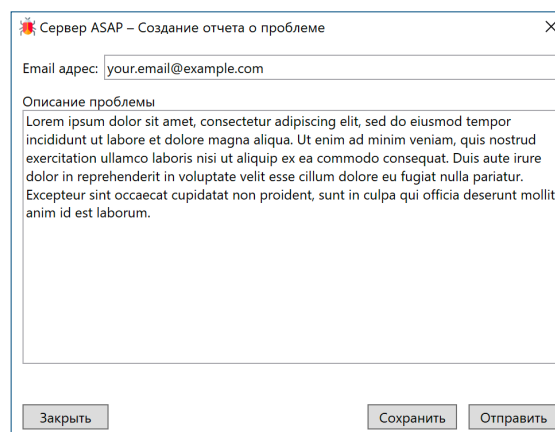
У лівій частині екрана розташований список зареєстрованих ліцензій. Для додавання нової ліцензії необхідно натиснути кнопку "Імпортувати" і вказати файл(и) ліцензії (з розширенням *.lic або *.license). Також передбачена можливість експорту ліцензій. Буква в першій колонці вказує тип ліцензії: Basic, Advanved або Demo.

У розділі "Управління сесіями" відображаються всі поточні сесії роботи з сервером. Сесія – це, в залежності від типу сервісу, активне підключення або сеанс роботи з сервером. Для примусового завершення сесії необхідно обрати в списку необхідну сесію та натиснути кнопку "Завершити".

У випадку сервісу "Інвентаризація" сесія – це сеанс роботи ТСД з сервером, тобто часовий проміжок між завантаженням даних на ТСД та вивантаженням оброблених даних до 1С. У випадку сервісу "Дисплей покупця" – це активне підключення дисплея до сервера ASAP.

Створення звіту про проблему

У разі виникнення проблем в роботі сервера існує можливість створення діагностичних звітів (рисунок 4.4). Для цього з меню Пуск необхідно запустити "ASAP Solutions ► Сервер ASAP ► Створення звіту про проблему" або натиснути на значок жука в нижньому правому куті "Конфігуратора".



Сервер ASAP – Создание отчета о проблеме

Email адрес:

Описание проблемы

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Закрити Сохранить Отправить

Рисунок 4.4 – Створення звіту про проблему

Для генерації звіту необхідно вказати адресу електронної пошти для зворотного зв'язку, а також детально описати проблему. Для відправки звіту про проблему слід натиснути на відповідну кнопку "Відправити". Після чого на вказану адресу електронної пошти буде надіслана копія звіту.

4.2.2 Сервіс "Інвентаризація"

Сервіс "Інвентаризація" призначений для виконання контролю товарного складу документів: інвентаризації, надходження, відвантаження тощо. Підтримується робота як з штучними (в тому числі адресне зберігання), так і з ваговими товарами.

4.2.2.1 Загальні відомості

У спрощеному вигляді задача сервісу зводиться до вивантаження даних з 1С на ТСД, власне проведення інвентаризації (в загальному випадку інвентаризація полягає в актуалізації фактичної кількості номенклатури) і завантаженні оброблених даних назад до 1С. Робота сервісу базується на документі інвентаризації – будь-якому документі 1С, що має табличну частину.

Вивантаження даних проводиться згідно з правилами, описаним в рамках конфігурації профіля інвентаризації. Профіль об'єднує групу налаштувань, призначених для певного типу задачі, наприклад, надходження товару, його відвантаження, проведення інвентаризації залишків тощо, та має власні параметри підключення до бази даних 1С.

У процесі вивантаження даних формується зведена таблиця, так звана "таблиця інвентаризації", загальний вигляд якої наведено на рисунку 4.5.

| Штрих-код | Облікова кількість | Фактична кількість | Реквізити | |
|-----------|--------------------|--------------------|-----------|-------|
| | | | | |
| | | | | |
| | | | | |

Рисунок 4.5 – Загальний вигляд таблиці інвентаризації

Таблиця інвентаризації будується на основі зв'язування табличної частини документа з регістром відомостей штрих-кодів (або кодів вагових товарів). Правила зв'язування задаються в рамках конфігурації профіля інвентаризації.

Проведення інвентаризації відбувається за допомогою прикладного програмного забезпечення для ТСД і зводиться до сканування штрих-кодів необхідної номенклатури, в тому числі і комірок у випадку використання адресного зберігання.

Порядок запису результуючих даних до 1С наведено у таблиці 4.1. Усі дії окрім запису даних інвентаризації опціональні, тобто їх виконання залежить від конфігурації профіля інвентаризації.

Таблиця 4.1 – Порядок запису результуючих даних

| Дія | Опис |
|--|---|
| Зняття документа з проведення | Перед початком запису виконується зняття документа з проведення, якщо він вже був проведений. |
| Запис даних інвентаризації | Оновлення значень фактичної кількості для рядків табличної частини документа інвентаризації. Також включає в себе оновлення значень комірок у випадку налаштованого заповнення комірок за фактом в опціях адресного зберігання. |
| Оновлення суми | Оновлення сум для кожного рядка табличної частини та загальної суми документа. |
| Обробка невідомих штрих-кодів | Виклик методу "ASAPIнвентаризацияОбработатьНеизвестны еШК" зовнішньої обробки для обробки невідомих штрих-кодів на стороні 1С. |
| Запис невідомих штрих-кодів в коментар | Запис невідомих штрих-кодів в коментар документа. У разі використання зовнішньої обробки штрих-коди, оброблені на стороні 1С, не потрапляють в коментар. |
| Виклик зовнішньої пост-обробки | Виклик методу "ASAPIнвентаризацияПостОбработка" зовнішньої обробки. |
| Оновлення статусу документа | Оновлення статусу документа |
| Проведення документа | Проведення документа в оперативному або неоперативне режимі. |

У разі використання запису в транзакції при виникненні помилки на будь-якому з етапів всі внесені до цього зміни буде скасовано. Інакше помилки будуть проігноровані і робота буде продовжена з подальшою нотифікацією переліку невдалих дій.

Якщо документ інвентаризації був змінений в період між вивантаженням даних на ТСД і завантаженням обробленої інформації назад до 1С, поведінка сервера не детермінований. Для коректної роботи системи документ, а також його таблична частина повинні залишатися незмінними під час проведення інвентаризації.

4.2.2.2 Конфігурація сервісу

Параметри сервісу об'єднані в профілі (рисунок 4.6). Кожен профіль представляє групу налаштувань, призначених для певного типу задачі та має власні параметри підключення до 1С.

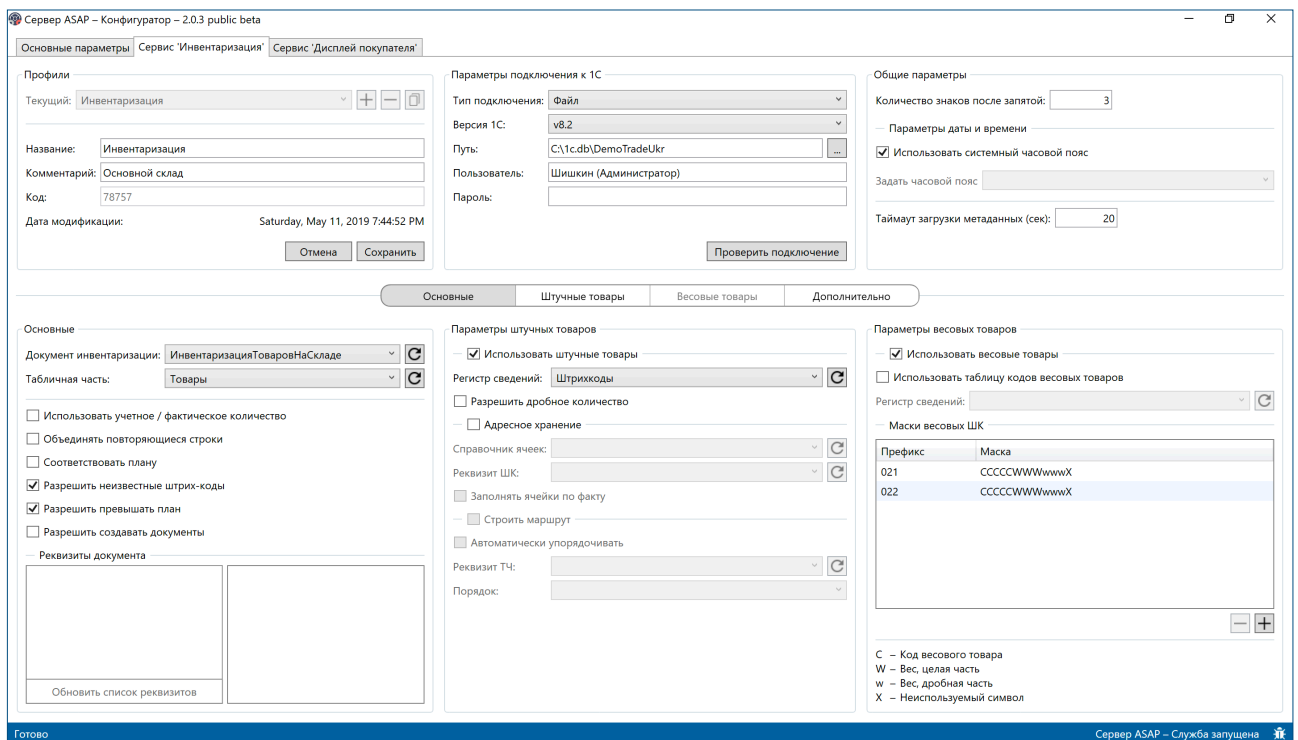


Рисунок 4.6 – Конфігурація профіля інвентаризації

Створення профіля

У вкладці "Сервіс 'Інвентаризація'" в групі "Профілі" слід натиснути кнопку "+" і вказати ім'я нового профіля. Профілю автоматично буде присвоєно унікальний код. Опціонально можна вказати коментар. Коментар буде

відображений на ТСД в списку доступних користувачеві профілів.

Налаштування підключення до 1С

В першу чергу необхідно обрати тип підключення: файлове або серверне. Для підключення до файлової версії 1С слід вказати наступні параметри: версію платформи 1С, шлях до бази даних, ім'я користувача і пароль.

Для підключення до серверної версії 1С потрібно задати наступні параметри: версію платформи 1С, хост, ім'я бази, ім'я користувача і пароль.

Загальні параметри

Для коректної роботи системи необхідно, щоб кількість знаків після коми в опціях профіля сервісу "Інвентаризація" відповідало аналогічним значенням в системі 1С.

Параметри дати та часу дозволяють вказати часовий пояс, що має використовуватися в сервері автоматизації. Часовий пояс, вказаний в профілі, повинен відповідати часовому поясу, що використовується в 1С.

Таймаут завантаження метаданих встановлює максимальний час в секундах, що може бути виділено на завантаження метаданих з 1С. У разі перевищення зазначеного таймаута активна операція буде примусово перервана.

Основні параметри

В першу чергу в лівій частині вікна необхідно вказати документ, на основі якого буде проводитися інвентаризація, та вказати його табличну частину. Далі слід встановити необхідні параметри проведення інвентаризації.

Параметр "Використовувати облікову / фактичну кількість" встановлює спосіб обробки значень кількості номенклатури для рядків табличної частини документа інвентаризації: в разі встановлення прапорця використовується роздільна облікова та фактична кількість, в зворотньому випадку – єдина. Так, наприклад, у випадку інвентаризації зазвичай використовується роздільна облікова та фактична кількість, а у випадку надходження – єдина.

Опція "Об'єднувати повторювані рядки" вказує, яким чином слід обробляти дублікати рядків при завантаженні даних на ТСД. Якщо опція встановлена, дублікати будуть об'єднані в один запис, а кількість буде підсумовано. Інакше документ з дублікатами записів не буде завантажено.

Параметр "Відповідати плану" визначає, чи припустимі відхилення від запланованого значення облікової кількості (або єдиної кількості у випадку, якщо роздільна кількість не використовується).

Прапорець "Дозволити невідомі штрих-коди" вказує, чи допускається обробка невідомих штрих-кодів.

Опція "Дозволити перевищувати план" дозволяє встановити обмеження на перевищення облікової кількості (або єдиної кількості у випадку, якщо роздільна кількість не використовується). При цьому недостачі допускаються.

Параметр "Дозволити створювати документи" вказує, чи можливо початок інвентаризації з порожнім новим документом. Документ створюється за допомогою виклику відповідного методу зовнішньої обробки.

У розділі "Реквізити документа" налаштовується перелік реквізитів, що відображаються на ТСД в списку документів.

Параметри штучних товарів

Для використання штучних товарів в першу чергу необхідно встановити відповідний прапорець в групі "Параметри штучних товарів". Після чого слід вказати реєстр відомостей штрих-кодів.

У разі використання адресного зберігання необхідно встановити прапорець "Адресне зберігання", вказати довідник комірок та реквізит комірки, що позначає штрих-код.

Параметр "Заповнювати комірки за фактом" встановлює режим запису документа, при якому значення комірок в табличній частині документа інвентаризації заповнюються фактичними значеннями, отриманими в рамках безпосереднього проведення інвентаризації.

Параметр "Будувати маршрут" вмикає режим маршруту при проведенні інвентаризації. В даному режимі ТСД вимагає дотримання порядку сканування комірок та товарів.

Параметри вагових товарів

Для використання вагових товарів необхідно встановити відповідний прапорець в групі "Параметри вагових товарів" в правій частині вікна. Можливе використання як окремого реєстра відомостей кодів вагових товарів, так і

загального реєстра відомостей штрих-кодів, зазначеного в параметрах штучних товарів.

Також необхідно вказати маски вагових штрих-кодів. Маска складається з префіксу і безпосередньо тіла маски. Префікс може містити будь-які символи. Тіло маски складається з коду товару, що позначається символом "С" і ваги, де "W" – ціла частина, а "w" – дробова. Символ "X" позначає символи, що не використовуються (наприклад, контрольну суму).

Конфігурація таблиці інвентаризації

Таблиця інвентаризації – це зведена таблиця, що будується в результаті об'єднання табличної частини документа інвентаризації та реєстра відомостей штрих-кодів (або кодів вагових товарів).

Для налаштування таблиці інвентаризації штучних товарів слід перейти на вкладку "Штучні товари" (рисунок 4.7). Після чого необхідно встановити відповідність між табличною частиною документа інвентаризації та реєстром відомостей в правій частині вікна конфігуратора, попередньо обравши необхідні реквізити табличної частини і реєстра відомостей у відповідних групах в лівій частині вікна.

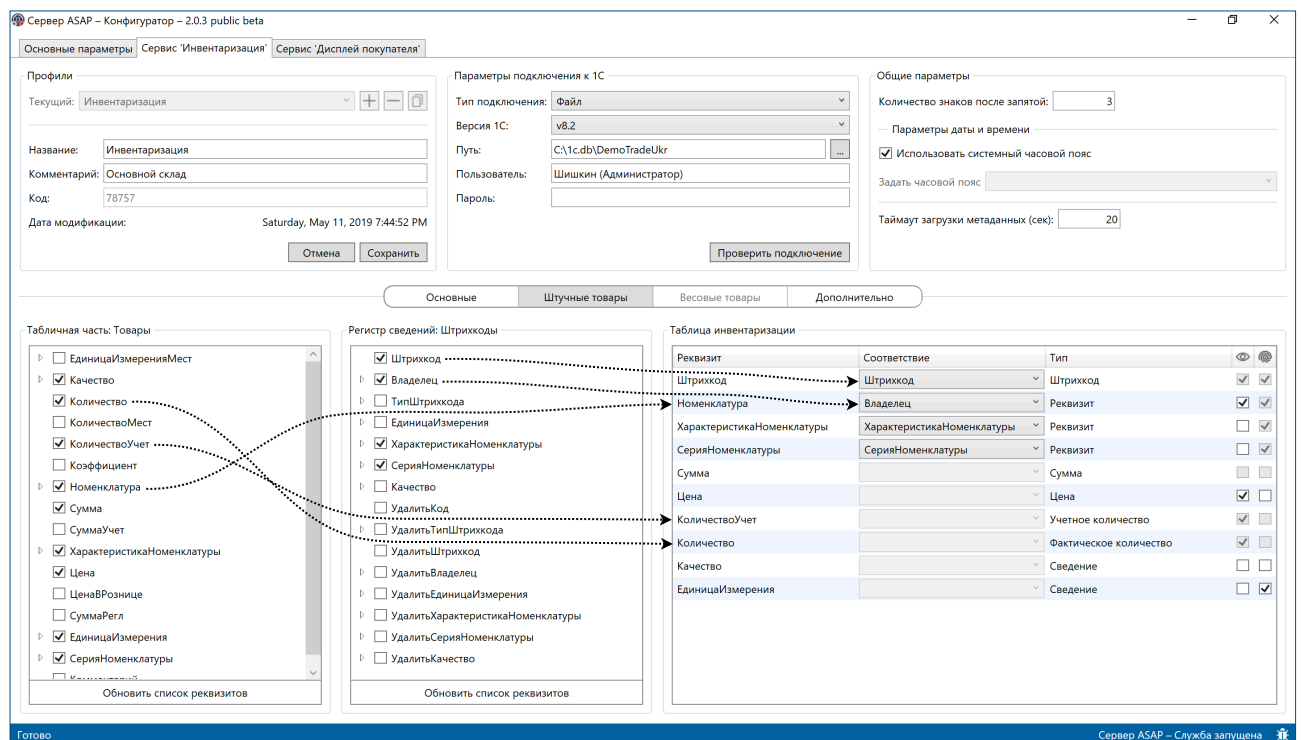


Рисунок 4.7 – Конфігурація штучних товарів

У випадку використання вагових товарів за умови використання таблиці кодів вагових товарів необхідна незалежна конфігурація відповідної таблиці інвентаризації.

Конфігурація таблиці інвентаризації вагових товарів виконується у вкладці "Вагові товари" аналогічно до таблиці інвентаризації штучних товарів.

4.2.2.3 Додаток для ТСД "ASAP – Інвентаризація"

Додаток "ASAP – Інвентаризація" є частиною програмного комплексу і може використовуватися виключно в зв'язці з сервером автоматизації.

Додаток призначений для безпосереднього проведення інвентаризації кінцевим користувачем відповідно до параметрів, встановлених у програмі "Конфігуратор".

Авторизація

Для початку роботи з додатком необхідно виконати процедуру авторизації. Для цього на основному екрані необхідно ввести ім'я користувача і пароль та натиснути "Увійти" (рисунок 4.8).

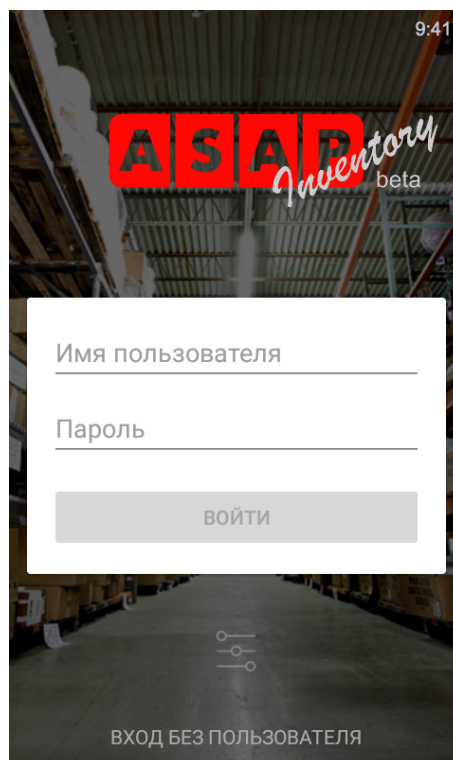


Рисунок 4.8 – Екран авторизації

Після успішної авторизації користувачеві буде показаний список доступних йому профілів.

Початок інвентаризації

Для початку інвентаризації (рисунки 4.9) слід обрати профіль зі списку, після чого необхідно обрати необхідний документ або вибрати пункт "Створити документ". Після підтвердження початку інвентаризації почнеться процес завантаження даних. Це може зайняти кілька хвилин.

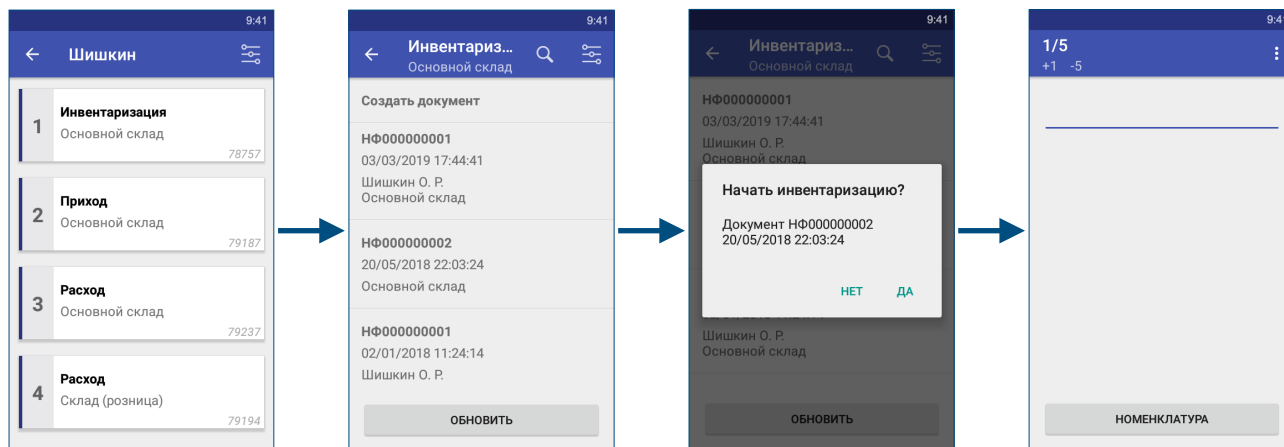


Рисунок 4.9 – Початок інвентаризації

Після успішного завантаження даних буде відкрито основне вікно інвентаризації.

Проведення інвентаризації

Процес проведення інвентаризації виконується шляхом сканування штрих-кодів на основному екрані інвентаризації (рисунки 4.10). В результаті сканування штрих-коду відображається основна інформація про товар.

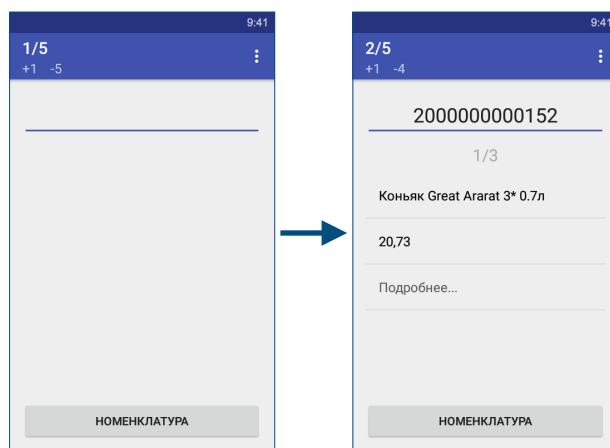


Рисунок 4.10 – Сканування товару

Кнопка "Детальніше" відкриє вікно з детальним описом відсканованого елемента з повним переліком реквізитів.

Для перегляду всіх завантажених з 1С даних необхідно натиснути кнопку "Номенклатура" внизу основного екрана інвентаризації. При натисканні на окремий елемент списку номенклатури буде відкрита детальна інформація про обраний елемент.

Завершення інвентаризації

Для завершення інвентаризації необхідно на основному екрані інвентаризації у випадаючому меню зверху справа обрати відповідний пункт. Після цього почнеться процес вивантаження даних до 1С. Це може зайняти кілька хвилин.

Також існує можливість скасування інвентаризації, що призведе до повної втрати всіх поточних даних інвентаризації.

4.2.2.4 Ліцензування сервісу

Сервіс "Інвентаризація" передбачає використання трьох типів ліцензій, ключові відмінності яких наведені у таблиці 4.2.

Таблиця 4.2 – Типи ліцензій сервісу "Інвентаризація"

| Функція | Demo | Basic | Advanced |
|---------------------------------|------|-------|----------|
| Запис даних | × | ✓ | ✓ |
| Створення нового документа | × | × | ✓ |
| Багатокористувацький режим | ✓ | × | ✓ |
| Адресне зберігання | ✓ | × | ✓ |
| Об'єднання неунікальних рядків | ✓ | × | ✓ |
| Використання зовнішньої обробки | × | × | ✓ |
| Використання WEB-сервісу | × | × | ✓ |
| Оновлення статусу документа | × | × | ✓ |
| Проведення документа | × | × | ✓ |

4.2.3 Сервіс "Дисплей покупця"

Сервіс "Дисплей покупця" призначений для організації інформаційного дисплея, здатного відображати інформацію, що надходить через RESTful API в форматі xml, а під час простою - для показу промо та рекламних матеріалів.

4.2.3.1 Загальні відомості

Сервіс "Дисплей покупця" може використовуватися як у зв'язці з програмним комплексом "1С: Підприємство 8", так і незалежно від нього.

У випадку інтеграції сервісу з 1С для організації інформаційного дисплея каси, для кожного пристрою встановлюється відповідність торговій точці, для якої призначений дисплей. Далі з 1С виконується API запит для передачі необхідних для поточної торгової точки даних. У пакет поставки входить клієнтська бібліотека для взаємодії з сервером автоматизації.

При використанні сервісу незалежно від 1С передача даних на дисплеї виконується за допомогою API запиту із зазначенням ідентифікатора пристрою.

Для коректної роботи системи між додатком "Дисплей покупця" і сервером необхідно постійне мережеве з'єднання. За відсутності з'єднання дисплей покупця продовжить працювати в режимі рекламного табло.

4.2.3.2 Конфігурація сервісу

При використанні сервісу у зв'язці з 1С необхідно сконфігурувати параметри підключення до бази даних 1С. Якщо ж сервіс використовується незалежно, даний етап конфігурації може бути пропущеним. Далі слід сформувати список використовуваних пристроїв та сконфігурувати промо-матеріали для відображення на дисплеях.

Налаштування підключення до 1С

Сервіс підтримує одночасну роботу з декількома екземплярами 1С. Параметри роботи з 1С об'єднані в іменовані групи, кожна з яких містить параметри підключення до 1С та опції довідника торгових точок.

В першу чергу необхідно обрати тип підключення: файлове або серверне, та налаштувати відповідні параметри. Для підключення до файлової версії 1С слід вказати версію платформи 1С, шлях до бази даних, ім'я користувача і пароль, а для серверної версії – версію платформи 1С, хост, ім'я бази, ім'я

користувача і пароль. Після чого потрібно обрати довідник торгових точок.

Керування пристроями

У правій частині вікна відображається перелік сконфігурованих пристроїв (рисунок 4.11). Він може бути як заповнений вручну, так і згенерований на основі існуючих ліцензій. Відмітка на початку рядка позначає наявність ліцензії для даного пристрою.

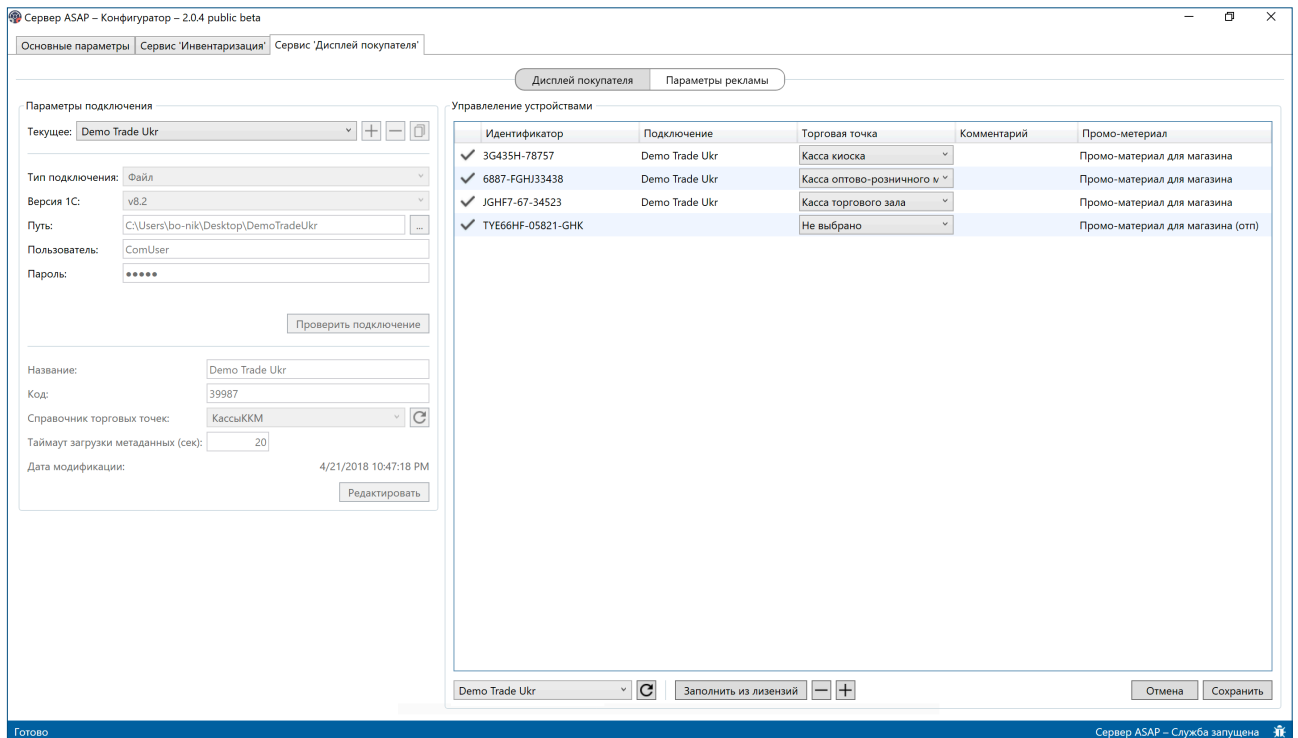


Рисунок 4.11 – Керування пристроями сервісу "Дисплей покупця"

У випадку інтеграції з 1С для конфігурації пристрою необхідно вказати відповідну до дисплея торгову точку. Для цього слід попередньо завантажити перелік доступних торгових точок з 1С, скориставшись кнопкою "оновити" біля активного підключення до 1С в лівій нижній частині області конфігурації пристроїв.

Конфігурація промо-матеріалів

Для конфігурації промо-матеріалів необхідно перейти на вкладку "Параметри реклами" (рисунок 4.12). Ліва область екрану містить елементи управління медіа файлами. У центральній частині розміщені елементи, призначені для створення і конфігурації промо-матеріалу. Права частина вікна містить інструменти зв'язування промо-матеріалів та пристроїв.

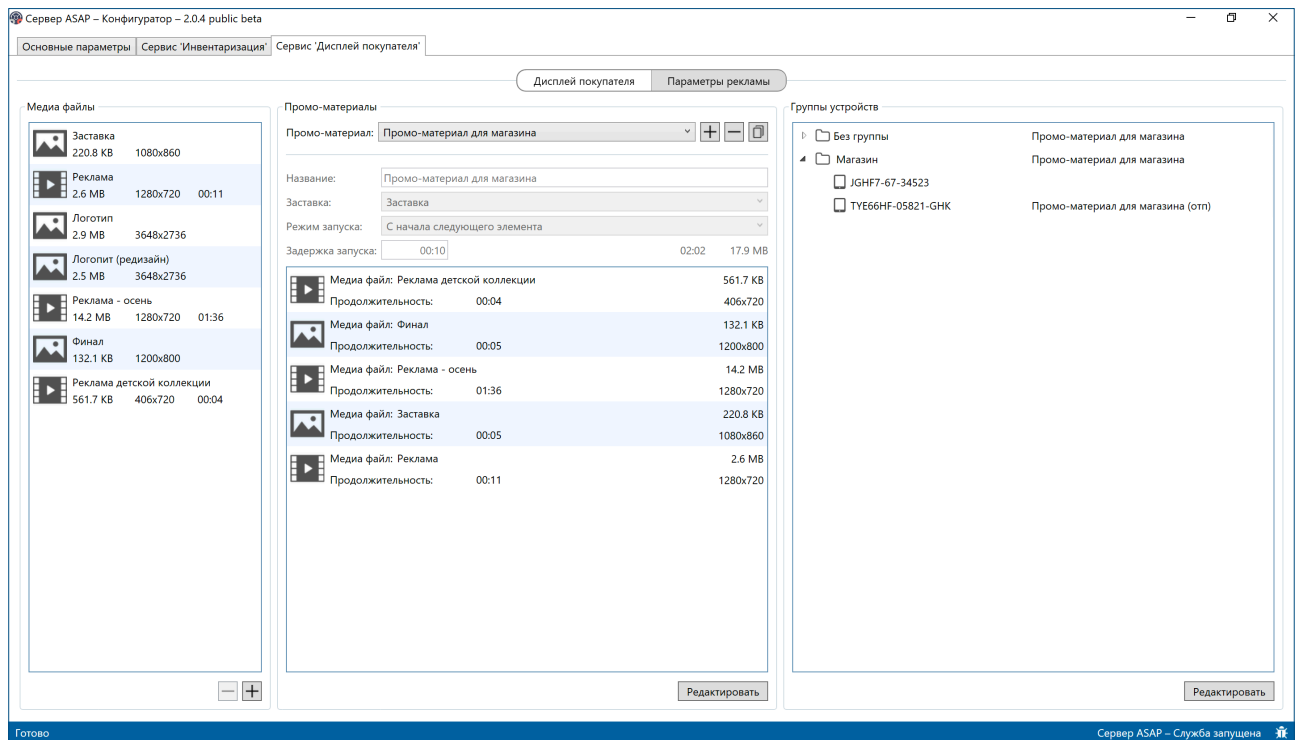


Рисунок 4.12 – Конфігурація параметрів відображення промо-матеріалів

Кожен промо матеріал складається з набору медіа файлів (фото або відео) та початкової заставки. У випадку, якщо встановлена ненульова затримка запуску, цей період часу буде відображатися заставка (наприклад, логотип).

Для додавання медіа до промо-матеріалу слід перетягнути необхідний елемент зі списку медіа файлів, розташованого в лівій частині екрана, в область редагування поточного промо-матеріалу. Для зображення слід вказати тривалість його демонстрації.

Область "Групи пристроїв" дозволяє об'єднувати дисплеї в групи та виконувати прив'язування промо-матеріалів як до всієї групи пристроїв, так і до окремих дисплеїв. У випадку, якщо для групи вказано промо-матеріал, а для пристрою, що знаходиться в цій групі, ні, то буде використана конфігурація групи. Якщо ж для окремого пристрою в групі параметри перевизначені, то вони матимуть більший пріоритет, ніж параметри групи.

4.2.3.3 Додаток "ASAP – Дисплей покупця"

Додаток "Дисплей покупця" є частиною програмного комплексу і може бути використано виключно в зв'язці з сервером автоматизації.

Додаток запускається автоматично під час запуску операційної системи

Android. У випадку якщо попередньо був налаштований промо-матеріал для даного пристрою та всі необхідні медіа файли були завантажені, почнеться показ промо-матеріалів.

Постійне мережеве з'єднання між додатком "Дисплей покупця" та сервером є обов'язковим, а за його відсутності дисплей покупця продовжить працювати виключно в режимі рекламного табло.

Відразу після запуску додаток виконує спробу з'єднання з сервером та не припиняє спроби до успішного з'єднання. При обриві з'єднання в верхньому правому кутку додатка відображається статус підключення.

Оновлення медіа файлів та інформації про правила їх відображення виконується автоматично відповідно до конфігурації інтервалу оновлення.

Для доступу до параметрів додатку слід натиснути й утримувати область у верхньому правому кутку екрану впродовж 5 секунд. Після цього буде відкрито вікно із запитом пароля. У разі успішного введення пароля відкриються параметри додатку (рисунок 4.13).

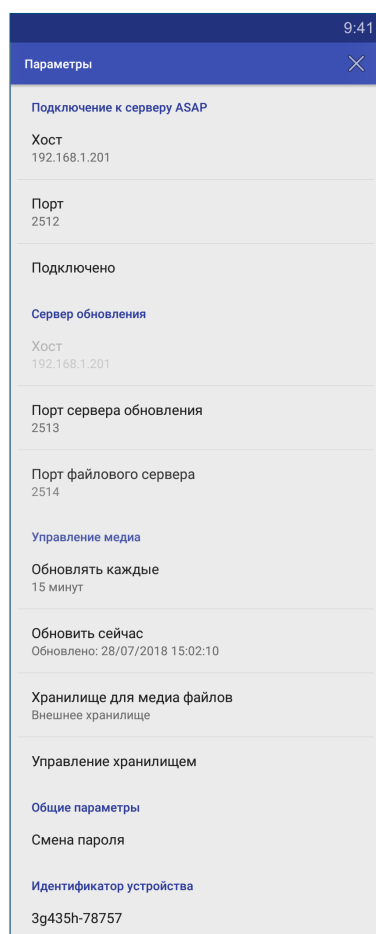


Рисунок 4.13 – Параметры додатку "Дисплей покупця"

Для мінімальної конфігурації додатку слід вказати лише адресу хоста, на якому запущений сервер автоматизації. Нижче відображається статус підключення або текст помилки в разі невдачі. Натискання на статус призведе до повторної спроби встановити з'єднання.

Параметр "Оновлювати кожні" дозволяє встановити інтервал перевірки наявності змін в конфігурації промо-матеріалу з боку сервера, та, за необхідності, завантаження нових медіа файлів.

Опція "Сховище медіа файлів" дозволяє вказати місце зберігання медіа файлів: внутрішня пам'ять пристрою або зовнішнє сховище (наприклад, SD накопичувач). Кнопка "Управління сховищем" відкриває вікно з поточним статусом завантажень.

Розділ "Загальні параметри" містить пункт "Зміна пароля", призначений для зміни пароля входу до конфігураційного режиму.

Параметр "Ідентифікатор пристрою" – це рядкове значення, що вказується користувачем та використовується сервером для ідентифікації пристроїв.

Висновки до розділу

В результаті опису та аналізу процесу впровадження та експлуатації програмного комплексу можна зробити наступні висновки:

- описано структуру розгорнутого програмного комплексу системи автоматизації торгівельних процесів, розглянуто середовище виконання кожного із компонентів програмного продукту та сформульовано вимоги до каналів зв'язку між елементами системи;
- описано механізм розгортання та інтеграції програмного забезпечення до існуючої інфраструктури та інформаційної системи підприємства торгівлі, сформульовано необхідні та достатні вимоги до інтерфейсу взаємодії сервера автоматизації та існуючої конфігурації 1С;
- розроблено інструкцію користувача системи автоматизації, що покроково описує процес конфігурації сервера автоматизації, а також механізм взаємодії користувача з мобільними додатками.

ВИСНОВКИ

В результаті проведеного магістерського дослідження можна зробити наступні висновки:

1. Виконано аналіз предметної області та виявлено проблему взаємодії між продуктами компанії 1С та стороннім торговельним обладнанням. Актуалізовано проблему інтеграції спеціалізованого торговельного обладнання до існуючих інформаційних систем підприємств та обґрунтовано необхідність створення уніфікованого механізму інтеграції, що дозволить мінімізувати ризики впливу на функціонування ключових бізнес-процесів організації.

2. Проаналізовано існуючі технічні засоби, виявлено, що жоден з існуючих програмних продуктів не дозволяє виконувати інтеграцію торговельного обладнання та платформи "1С:Підприємство 8" без потреби внесення змін до конфігурації.

3. Сформульовано задачу розробки універсального підходу до інтеграції спеціалізованого торговельного обладнання з метою автоматизації бізнес-процесів підприємства.

4. Проведено аналіз типової інформаційної системи торговельного підприємства на базі програмного комплексу "1С: Підприємство 8" та спроектовано концептуальну модель уніфікованого механізму побудови міжсистемної взаємодії платформи "1С: Підприємство 8" та спеціалізованого обладнання (рисунок 2.3).

5. На основі теоретично обґрунтованого уніфікованого механізму інтеграції спеціалізованого обладнання було спроектовано та розроблено прикладне програмне забезпечення для автоматизації окремих бізнес-процесів товарообігу підприємства торгівлі з використанням терміналів збору даних та дисплеїв покупця, зокрема процесів інвентаризації та роздрібно́ї торгівлі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Randy Abernethy. The Programmer's Guide to Apache Thrift. – Manning Publications Company, 2015. – 500 с.
2. Rob Miles. C# Programming Yellow Book. – 2016. – 417 с.
3. Zigurd Mednieks, Laird Dornin, G. Blake Meike, Masumi Nakamura. Programming Android: Java Programming for the New Generation of Mobile Devices (second edition). – O'Reilly Media, 2012. – 566 с.
4. А. П. Габец, Д.И. Гончаров. 1С:Предприятие 8.2. Простые примеры разработки – М. 1С-Паблишинг, 2017. – 218 с.
5. Андерсон К. Основы Windows Presentation Foundation. – СПб.: БХВ-Петербург, 2008. – 432 с.
6. Баричев С. Г., Гончаров В. В., Серов Р. Е. Стандарт AES. Алгоритм Rijdael. – 3-е изд. – М.: Диалог-МИФИ, 2011. – 176 с.
7. Бойко В. Проектирование баз данных информационных систем / Бойко В. – М.: Финансы и статистика, 1989. – 838 с.
8. Габец А.П. Реализация прикладных задач в системе “1С:Предприятие 8.2” / А. П. Габец, Д. В. Козырев, Д. С. Кухлевский, Е. Ю. Хрусталева – М.: 1С-Паблишинг, 2013. – 714 с.
9. Гайдаржи В. І. Основи проектування та використання баз даних: Навч. посібник. – 2-ге вид., виправл. і допов. / В. І. Гайдаржи, О. А. Дацюк – К.: ІВЦ “Видавництво “Політехніка”, ТОВ “Фірма “Періодика”, 2004. – 256 с.
10. Голощапов А. Google Android: программирование для мобильных устройств. – СПб.: БХВ-Петербург, 2010. – 448 с.
11. Дейт К. Введение в систему баз данных / Дейт К. – М.: Мир, 1998. – 532 с.
12. Джудит С. Практическое руководство по SQL / С. Джудит – М.: Символ-Плюс, 2007. – 715 с.
13. Ибе О. Сети и удаленный доступ. Протоколы, проблемы, решения / О. Ибе. – М.: ДМК Пресс, 2002. – 336 с.

14. Калбертсон Роберт, Браун Крис, Кобб Гэри. Быстрое тестирование. – М.: Вильямс, 2002. – 374 с.
15. Канер Кем, Фолк Джек, Нгуен Енг Кек. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений. – Киев: ДиаСофт, 2001. – 544 с.
16. Коутинхо С. Введение в теорию чисел. Алгоритм RSA = The Mathematics of Ciphers: Number Theory and RSA Cryptography. – М.: Постмаркет, 2001. – 328 с.
17. Мартин Р. Принципы, паттерны и методики гибкой разработки на языке C#. – 2016. – 573 с.
18. Радченко М. Г., Хрусталева Е. Ю. 1С:Предприятие 8.3. Практическое пособие разработчика. – М. 2019. – 387 с.
19. Резник С., Крейн Р., Боуэн К. Основы Windows Communication Foundation для .NET Framework 3.5 – М.: ДМК Пресс, 2008. – 387 с.
20. Филиппов Е. В. Настольная книга 1С: Эксперта по технологическим вопросам – М.: 1С-Паблишинг, 2010. – 247 с.
21. Хорстманн К. С. Java SE 8. Вводный курс. – М.: «Вильямс», 2014. – 208 с.
22. Хрусталева Е. Ю. Технологии интеграции 1С:Предприятия 8.3 – М.: 1С-Паблишинг, 2020. – 415 с.

ДОДАТОК А. ТЕКСТИ ПРОГРАМНОГО КОДУ

```
// InvenroryService.cs ////////////////////////////////////////

using System;
using System.Collections.Generic;
using ASAP.Api.Thrift;
using ASAP.Api.Thrift.Inventory;
using System.Diagnostics;
using ASAP.Common;
using ASAP.Logging;
using System.Linq;
using ASAP.Supporting.Extensions;
using ASAP.Provisioning;

namespace ASAP.Services.Inventory {

    public class Handler : InventoryService.Iface {

        private ILog logger = LogManager.GetLogger("Inventory.Handler");

        ////////////////////////////////////////

        private List<InventoryDocument> pendingDocuments = new
List<InventoryDocument>();

        private class InventoryDocumentComparer :
IEqualityComparer<InventoryDocument> {

            public bool Equals(InventoryDocument lhs, InventoryDocument rhs) {
```

```

return lhs.Code == rhs.Code && lhs.Date == rhs.Date;
}

```

```

public int GetHashCode(InventoryDocument document) {
    return document.Code.GetHashCode() & document.Date.GetHashCode();
}
}

```

```

// InventoryService.Iface //////////////////////////////////////

```

```

    public GetInventoryDocumentsResponse
    GetInventoryDocuments(ConnectionToken token, string profile,
    InventoryDocumentFilter filter, int timeout) {
        logger.Info($"GetInventoryDocuments: Device {token.Device}: Accept
incomming connection, profile {profile}");
        var watch = Stopwatch.StartNew();

        GetInventoryDocumentsResponse response;
        try {
            response = ExecuteGetInventoryDocuments(token, profile, filter, timeout);
        } catch (Exception ex) {
            logger.Warn($"GetInventoryDocuments: Device {token.Device}: Exception
occured", ex);
            response = Helpers.Response.GetInventoryDocuments.Fail(ex);
        }

        watch.Stop();
        long elapsedMs = watch.ElapsedMilliseconds;

        logger.Info($"GetInventoryDocuments: Device {token.Device}: Processing
finished in {elapsedMs} ms. " +

```

```

        $"Responde {response.Status.Code}, Message:
'{response.Status.Message}");
        RemoteLogger.Post(
            $" {InventoryServiceConstants.SERVICE_NAME} -
GetInventoryDocuments",
            token.Device,
            $"Responde {response.Status.Code}, Message: '{response.Status.Message}'"
        );

        return response;
    }

```

```

    public FetchInventoryDataResponse FetchInventoryData(ConnectionToken
token, InventorySession session, int timeout) {
        logger.Info($"FetchInventoryData: Device {token.Device}: Accept
incomming connection, " +
            $"profile {session.Profile}, document {session.Document.Code}
{session.Document.Date}");
        var watch = Stopwatch.StartNew();

        FetchInventoryDataResponse response;
        try {
            response = ExecuteFetchInventoryData(token, session, timeout);
        } catch (Exception ex) {
            logger.Warn($"FetchInventoryData: Device {token.Device}: Exception
occured", ex);
            response = Helpers.Response.FetchInventoryData.Fail(ex);
        }

        if (response.Status.Code == StatusCode.FAIL || response.Status.Code ==
StatusCode.UNAUTHORIZED) {

```

```

    if (response.Status.__isset.session) {
        try {
            SessionsManager.Shared.DestroySession(response.Status.Session);
        } catch (Exception ex) {
            logger.Warn($"FetchInventoryData: Device {token.Device}: Failed to
destroy session", ex);
        }
        response.Status.Session = null;
        response.Status.__isset.session = false;
    }
}

watch.Stop();
long elapsedMs = watch.ElapsedMilliseconds;

logger.Info($"FetchInventoryData: Device {token.Device}: Processing
finished in {elapsedMs} ms. " +
    $"Responde {response.Status.Code}, Message:
'{response.Status.Message}");
RemoteLogger.Post(
    $" {InventoryServiceConstants.SERVICE_NAME} - FetchInventoryData",
    token.Device,
    $"Responde {response.Status.Code}, Message: '{response.Status.Message}'"
);

return response;
}

public PushInventoryDataResponse PushInventoryData(ConnectionToken
token, InventorySession session, List<NomenclatureItem> nomenclature, int timeout)
{

```

```

        logger.Info($"PushInventoryData: Device {token.Device}: Accept
incoming connection, " +
            $"profile {session.Profile}, document {session.Document.Code}
{session.Document.Date}");
        var watch = Stopwatch.StartNew();

        PushInventoryDataResponse response;
        try {
            response = ExecutePushInventoryData(token, session, nomenclature,
timeout);
        } catch (Exception ex) {
            logger.Warn($"PushInventoryData: Device {token.Device}: Exception
occured", ex);
            response = Helpers.Response.PushInventoryData.Fail(ex);
        }

        lock (pendingDocuments) {
            string date = (new
DateTime()).FromUnixTimestamp(session.Document.Date).ToString("yyyy.MM.dd
HH:mm:ss");

            logger.Info($"PushInventoryData: Device { token.Device}: Unlock
document {session.Document.Code} {date}");

            pendingDocuments.RemoveAll(doc => session.Document.Code ==
doc.Code && session.Document.Date == doc.Date);
        }

        if (response.Status.Code != StatusCode.FAIL && response.Status.Code !=
StatusCode.UNAUTHORIZED) {
            try {
                SessionsManager.Shared.DestroySession(token);
            } catch (Exception ex) {

```

```

        logger.Warn($"PushInventoryData: Device {token.Device}: Failed to
destroy session", ex);

        response.Status.Code = StatusCode.WARNING;
        string message = $"Не удалось завершить сессию: {ex.Message}";
        response.Status.Message = response.Status.__isset.message ?
        $"{Environment.NewLine}{message}" : message;
    }
}

watch.Stop();
long elapsedMs = watch.ElapsedMilliseconds;

logger.Info($"PushInventoryData: Device {token.Device}: Processing
finished in {elapsedMs} ms. " +
    $"Responde {response.Status.Code}, Message:
'{response.Status.Message}'");
RemoteLogger.Post(
    $"{InventoryServiceConstants.SERVICE_NAME} - PushInventoryData",
    token.Device,
    $"Responde {response.Status.Code}, Message: '{response.Status.Message}'"
);

return response;
}

public DismissInventoryResponse DismissInventory(ConnectionToken token,
int timeout) {
    logger.Info($"DismissInventory: Device {token.Device}: Accept incoming
connection");
    var watch = Stopwatch.StartNew();

```

```

DismissInventoryResponse response;
try {
    response = ExecuteDismissInventory(token, timeout);
} catch (Exception ex) {
    logger.Warn($"DismissInventory: Device {token.Device}: Exception
occured", ex);
    response = Helpers.Response.DismissInventory.Fail(ex);
}

watch.Stop();
long elapsedMs = watch.ElapsedMilliseconds;

logger.Info($"DismissInventory: Device {token.Device}: Processing finished
in {elapsedMs} ms. " +
    $"Responde {response.Status.Code}, Message:
'{response.Status.Message}");
RemoteLogger.Post(
    $"{InventoryServiceConstants.SERVICE_NAME} - DismissInventory",
    token.Device,
    $"Responde {response.Status.Code}, Message: '{response.Status.Message}'"
);

return response;
}

public ResetInventoryResponse ResetInventory(ConnectionToken token, int
timeout) {
    logger.Info($"ResetInventory: Device {token.Device}: Accept incomming
connection");
    var watch = Stopwatch.StartNew();

```

```

ResetInventoryResponse response;
try {
    response = ExecuteResetInventory(token, timeout);
} catch (Exception ex) {
    logger.Warn($"ResetInventory: Device {token.Device}: Exception occurred",
ex);

    response = Helpers.Response.ResetInventory.Fail(ex);
}

watch.Stop();
long elapsedMs = watch.ElapsedMilliseconds;

logger.Info($"ResetInventory: Device {token.Device}: Processing finished in
{elapsedMs} ms. " +
    $"Responde {response.Status.Code}, Message:
'{response.Status.Message}");
RemoteLogger.Post(
    $" {InventoryServiceConstants.SERVICE_NAME} - ResetInventory",
    token.Device,
    $"Responde {response.Status.Code}, Message: '{response.Status.Message}'"
);

return response;
}

public TestConnectionResponse TestConnection(ConnectionToken token, int
timeout) {
    logger.Info($"TestConnection: Device {token.Device}: Accept incomming
connection");
    var watch = Stopwatch.StartNew();

```



```

TestConnectionResponse response;
try {
    response = ExecuteTestConnection(token, timeout);
} catch (Exception ex) {
    logger.Warn($"TestConnection: Device {token.Device}: Exception
occured", ex);
    response = Helpers.Response.TestConnection.Fail(ex);
}

watch.Stop();
long elapsedMs = watch.ElapsedMilliseconds;

logger.Info($"TestConnection: Device {token.Device}: Processing finished
in {elapsedMs} ms. " +
    $"Responde {response.Status.Code}, Message:
'{response.Status.Message}");
RemoteLogger.Post(
    $"{InventoryServiceConstants.SERVICE_NAME} - TestConnection",
    token.Device,
    $"Responde {response.Status.Code}, Message: '{response.Status.Message}'"
);

return response;
}

```

// Implementations //////////////////////////////////////

```

private GetInventoryDocumentsResponse
ExecuteGetInventoryDocuments(ConnectionToken token, string profile,
InventoryDocumentFilter filter, int timeout) {
    var decision = ConnectionManager.Shared.AcceptConnection(

```

```

        token, InventoryServiceConstants.SERVICE_NAME,
        new ConnectionManager.ConnectionOptions() {
            Tag = profile
        });
        if (!decision.Accepted) {
            logger.Info($"ExecuteGetInventoryDocuments: Device {token.Device}:
Access denied for incoming connection");
            return
Helpers.Response.GetInventoryDocuments.Unauthorized(decision.Message);
        }

        GetInventoryDocumentsResponse response = null;
        try {
            Helpers.Executor.Execute((client) => {
                response = client.GetInventoryDocuments(token, profile, filter, timeout);
            }, timeout, logger);
        } catch (Exception ex) {
            logger.Warn($"ExecuteGetInventoryDocuments: Device {token.Device}:
Execution failed", ex);
            return Helpers.Response.GetInventoryDocuments.Fail(ex);
        }

        return response;
    }

    private FetchInventoryDataResponse
ExecuteFetchInventoryData(ConnectionToken token, InventorySession
inventorySession, int timeout) {
        var decision = ConnectionManager.Shared.AcceptConnection(
            token, InventoryServiceConstants.SERVICE_NAME,
            new ConnectionManager.ConnectionOptions() {

```

```

        Tag = inventorySession.Profile
    });
    if (!decision.Accepted) {
        logger.Info($"ExecuteFetchInventoryData: Device {token.Device}: Access
denied for incoming connection");
        return
Helpers.Response.FetchInventoryData.Unauthorized(decision.Message);
    }

    Session session;
    try {
        session = SessionsManager.Shared.CreateSession(token);
    } catch (Exception ex) {
        logger.Warn($"ExecuteFetchInventoryData: Device {token.Device}: Failed
to create session", ex);
        return Helpers.Response.FetchInventoryData.Unauthorized(ex.Message);
    }

    FetchInventoryDataResponse response = null;
    try {
        Helpers.Executor.Execute((client) => {
            response = client.FetchInventoryData(token, inventorySession, timeout);
        }, timeout, logger);
    } catch (Exception ex) {
        logger.Warn($"ExecuteFetchInventoryData: Device {token.Device}:
Execution failed", ex);
        return Helpers.Response.FetchInventoryData.Fail(ex);
    }

    response.Status.Session = session.Id;
    return response;

```

```

    }

    private PushInventoryDataResponse
ExecutePushInventoryData(ConnectionToken token, InventorySession session,
List<NomenclatureItem> nomenclature, int timeout) {
    var decision = ConnectionManager.Shared.AcceptConnection(
        token, InventoryServiceConstants.SERVICE_NAME,
        new ConnectionManager.ConnectionOptions() {
            RequireSessionExistence = true,
            Tag = session.Profile
        });
    if (!decision.Accepted) {
        logger.Info($"ExecutePushInventoryData: Device {token.Device}: Access
denied for incoming connection");
        return
Helpers.Response.PushInventoryData.Unauthorized(decision.Message);
    }

    lock (pendingDocuments) {
        string date = (new
DateTime()).FromUnixTimestamp(session.Document.Date).ToString("yyyy.MM.dd
HH:mm:ss");
        if (pendingDocuments.Contains(session.Document, new
InventoryDocumentComparer())) {
            logger.Info($"ExecutePushInventoryData: Device {token.Device}:
Document {session.Document.Code} {date} is busy");
            return Helpers.Response.PushInventoryData.Fail("Документ находится в
работе");
        }
        logger.Info($"ExecutePushInventoryData: Device { token.Device}: Lock
document {session.Document.Code} {date}");
    }
}

```

```

    pendingDocuments.Add(session.Document);
}

PushInventoryDataResponse response = null;
try {
    Helpers.Executor.Execute((client) => {
        response = client.PushInventoryData(token, session, nomenclature,
timeout);
    }, timeout, logger);
} catch (Exception ex) {
    logger.Warn($"ExecutePushInventoryData: Device {token.Device}:
Execution failed", ex);
    return Helpers.Response.PushInventoryData.Fail(ex);
}

return response;
}

private DismissInventoryResponse
ExecuteDismissInventory(ConnectionToken token, int timeout) {
    var decision = ConnectionManager.Shared.AcceptConnection(token,
InventoryServiceConstants.SERVICE_NAME,
new ConnectionManager.ConnectionOptions() {
    RequireSessionExistence = true
});
    if (!decision.Accepted) {
        logger.Info($"ExecuteDismissInventory: Device {token.Device}: Access
denied for incomming connection");
        return Helpers.Response.DismissInventory.Unauthorized(decision.Message);
    }
}

```

```

    try {
        SessionsManager.Shared.DestroySession(token);
    } catch (Exception ex) {
        logger.Warn($"ExecuteDismissInventory: Device {token.Device}: Failed to
destroy session", ex);
        return Helpers.Response.DismissInventory.Warning($"Не удалось
завершить сессию: {ex.Message}");
    }

    return Helpers.Response.DismissInventory.Ok();
}

private ResetInventoryResponse ExecuteResetInventory(ConnectionToken
token, int timeout) {
    var decision = ConnectionManager.Shared.AcceptConnection(token,
InventoryServiceConstants.SERVICE_NAME,
new ConnectionManager.ConnectionOptions() {
    RequireAuthorization = false,
    RequireAccessRulesVerification = false
});
    if (!decision.Accepted) {
        logger.Info($"ExecuteResetInventory: Device {token.Device}: Access
denied for incoming connection");
        return Helpers.Response.ResetInventory.Unauthorized(decision.Message);
    }

    try {
        SessionsManager.Shared.DestroySession(token);
    } catch (Exception ex) {
        logger.Warn($"ExecuteResetInventory: Device {token.Device}: Failed to
destroy session", ex);

```

```

        return Helpers.Response.ResetInventory.Warning($"Не удалось завершить
сессию: {ex.Message}");
    }
    return Helpers.Response.ResetInventory.Ok();
}
private TestConnectionResponse ExecuteTestConnection(ConnectionToken
token, int timeout) {
    var decision = ConnectionManager.Shared.AcceptConnection(token,
        InventoryServiceConstants.SERVICE_NAME,
        new ConnectionManager.ConnectionOptions() {
            RequireAuthorization = false
        });
    if (!decision.Accepted) {
        logger.Info($"ExecuteTestConnection: Device {token.Device}: Access
denied for incoming connection");
        return Helpers.Response.TestConnection.Unauthorized(decision.Message);
    }
    TestConnectionResponse response = null;
    try {
        Helpers.Executor.Execute((client) => {
            response = client.TestConnection(token, timeout);
        }, timeout, logger);
    } catch (Exception ex) {
        logger.Warn($"ExecuteTestConnection: Device {token.Device}: Execution
failed", ex);
        return Helpers.Response.TestConnection.Fail(ex);
    }
    return response;
}
}
}
}

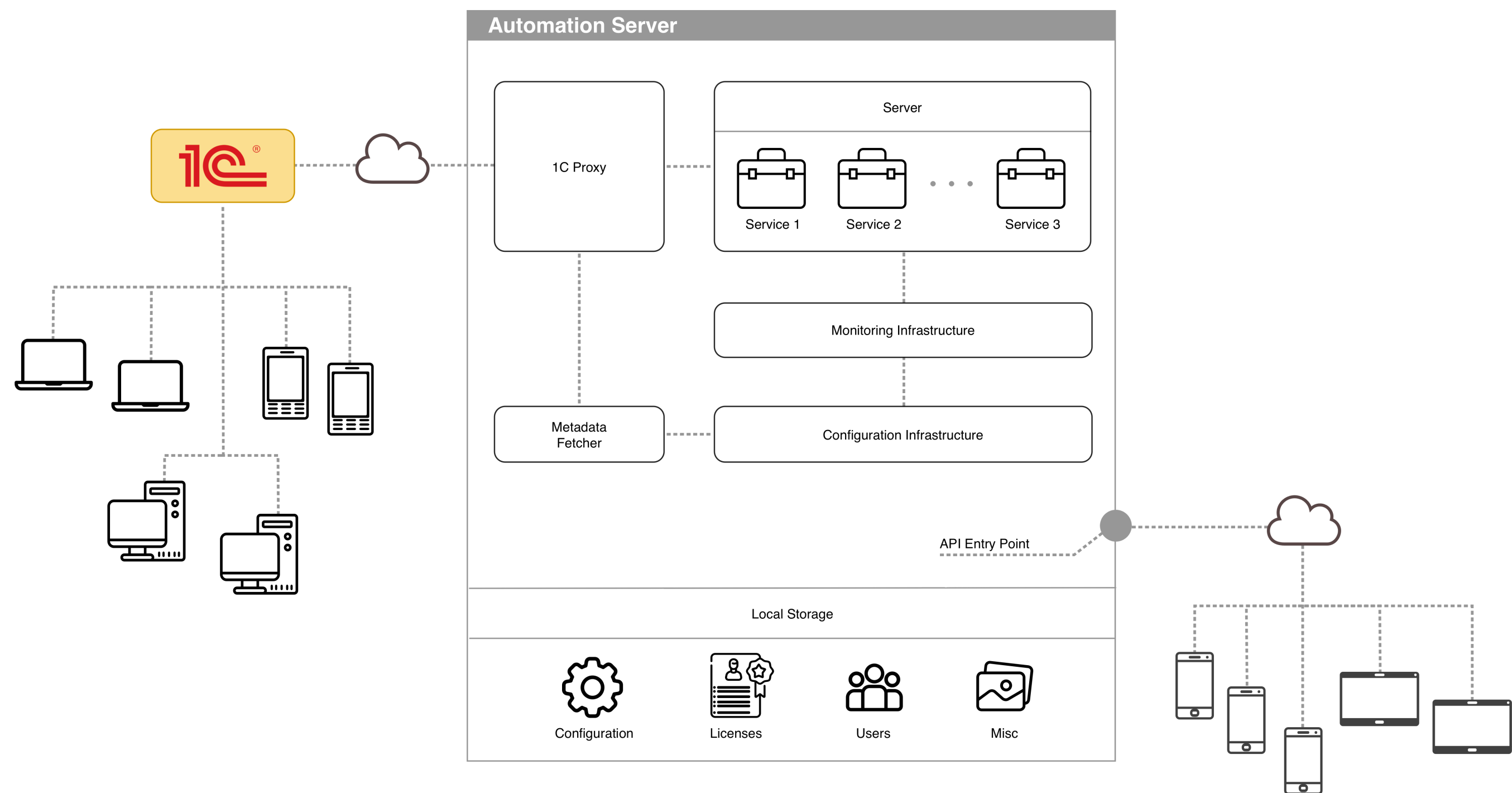
```

ДОДАТОК Б. ГРАФІЧНИЙ МАТЕРІАЛ

На плакаті "Концептуальна архітектура сервера автоматизації" зображено деталізовану модель уніфікованого механізму інтеграції спеціалізованого торговельного обладнання до інформаційної системи підприємства на базі програмної платформи "1С: Підприємство 8".

Плакат "Структурна схема компонентів сервера автоматизації" відображає узагальнену модель взаємодії компонентів прикладного рішення для інтеграції мобільних терміналів збору даних та дисплеїв покупця, побудованого на основі розробленого уніфікованого механізму.

КОНЦЕПТУАЛЬНА АРХІТЕКТУРА СЕРВЕРА АВТОМАТИЗАЦІЇ



Демонстраційний плакат до магістерської дисертації

Концептуальна архітектура сервера автоматизації

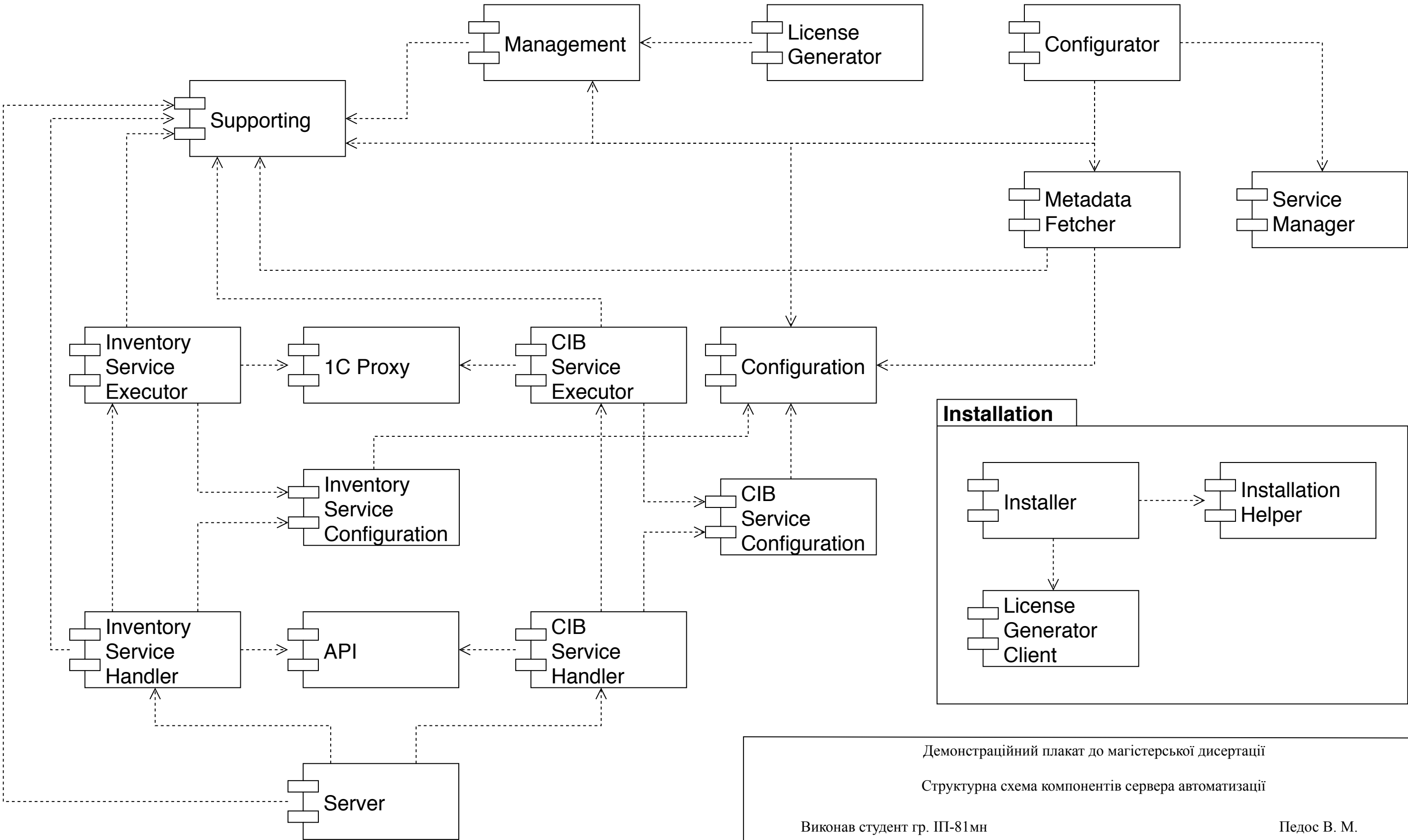
Виконав студент гр. ПІ-81мн

Педос В. М.

Керівник

Ліщук К. І.

СТРУКТУРНА СХЕМА КОМПОНЕНТІВ СЕРВЕРА АВТОМАТИЗАЦІЇ



Демонстраційний плакат до магістерської дисертації

Структурна схема компонентів сервера автоматизації

Виконав студент гр. ІІ-81мн

Педос В. М.

Керівник

Ліщук К. І.